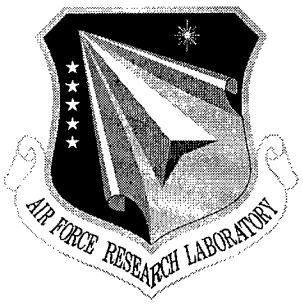
AFRL-IF-RS-TR-1999-53
Final Technical Report
March 1999



A KNOWLEDGE-BASED INTERFERENCE REJECTION SCHEME FOR DIRECT-SEQUENCE SPREAD-SPECTRUM SYSTEMS

Syracuse University

P. K. Varshney, D. D. Weiner, S. Hamid Nawab, I. Demirkiran, V. N. S. Samarasooriya, and Ramamurthy Mani

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

**AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK**

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-1999-53 has been reviewed and is approved for publication.

APPROVED: Stephen C. Tyler

STEPHEN C. TYLER
Project Engineer

FOR THE DIRECTOR:

WARREN H. DEBANY, JR., Technical Advisor
Information Grid Division
Information Directorate

If your address has changed or if you wish to be removed from the Air Force Research Laboratory Rome Research Site mailing list, or if the addressee is no longer employed by your organization, please notify AFRL/IFGC, 525 Brooks Road, Rome, NY 13441-4505. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 199	Final Jun 95 - Mar 98

4. TITLE AND SUBTITLE
A KNOWLEDGE-BASED INTERFERENCE REJECTION SCHEME FOR DIRECT-SEQUENCE SPREAD-SPECTRUM SYSTEMS

5. FUNDING NUMBERS
C - F30602-95-C-0204
PE - 62702F
PR - 4519
TA - 42
WU - 90

6. AUTHOR(S)
Syracuse University: P. K. Varshney, D. D. Weiner, I. Kemirkiran, and V. N. S. Samarasooriya
Boston University: S. Hamid Nawab, and Ramamurthy Mani

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Prime Contractor:
Syracuse University
Department of Electrical Engineering and Computer Sciences
Syracuse NY 13244

Subcontractor:
Boston University
Department of Electrical and Computer Engineering
Boston MA 02215

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory/IFGC
525 Brooks Road
Rome NY 13441-4505

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-1999-53

11. SUPPLEMENTARY NOTES
Air Force Research Laboratory Project Engineer: Stephen C. Tyler/IFGC/(315) 330-3618

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Spread-spectrum signals are used widely in military and commercial communication systems due to their interference rejection capability and their lower probability of interception. In military applications the effects of intentional interference (jamming) are mitigated by the processing gain of the spread-spectrum system. In many spread-spectrum systems, processing gain alone is not sufficient to achieve satisfactory system performance and additional interference rejection techniques need to be employed. The interference suppression circuit is placed prior to the spectrum despreader with the goal of reducing the jammer/interferer energy to an adequately low level that can be handled by the system processing gain.

We have presented a novel knowledge-based interference cancellation scheme for direct sequence spread-spectrum systems. This innovative approach utilizes:

* IPUS, an expert system for the Integrated processing and Understanding of signals, to monitor the communication signal environment in order to determine the parameters of interfering signals within a pre-specified accuracy, and
* Expert system rules to select from a library of preselected techniques, suitable interference rejection schemes based upon the knowledge obtained from monitoring the signal environment.

The effectiveness of this novel interference rejection capability is demonstrated by considering a number of interference scenarios and using the software package SPW©, a time-domain Signal Processing Worksystem.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Communications, Spread Spectrum	104
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Executive Summary

With the ever increasing signal density of the available electro-magnetic spectrum, the desired signal of one channel or system becomes the undesired signal or interference of another. When the number and nature of interference signals, including their frequency spectra or waveforms, are not known a priori, communications becomes quite difficult. In military communication environments, the interference that affects the communication could also be intentional and hostile. Another problem is that transmitted information may be intercepted by unintentional users. Therefore, additional signal processing techniques are required in communication systems both to mitigate the effects of intentional interference and to lower the probability of interception. Advanced modulation schemes employing spread-spectrum and coding techniques are commonly used in communication systems to decrease their vulnerability to interference and to lower their probability of interception.

In this study we focus on interference suppression in spread-spectrum communication systems. An effective technique used in spread-spectrum systems to minimize the effects of jamming is to pseudo-randomly distribute the information to be transmitted over a range of parameters (time, frequency and phase). Jamming protection is achieved because the jammer does not know the pseudo-random pattern and must distribute its limited resources (power) over many alternatives. This pseudo-random distribution of information also serves to conceal the information and thus prevents it from being intercepted. The inherent interference immunity displayed by a spread-spectrum system is not sufficient to combat all types of interference encountered in a communication environment. Additional

interference mitigation techniques are, therefore, needed to ensure effective and secure information transfer.

A significant number of interference cancellation schemes have been introduced in the literature. While these schemes are able to provide effective cancellation for specific types of interference, no single scheme is able to suppress all types of interference that is encountered by the spread-spectrum system. We present a novel knowledge-based interference cancellation scheme for direct-sequence spread-spectrum systems. This innovative approach utilizes

- *IPUS*, an expert system for the Integrated Processing and Understanding of Signals, to monitor the communication signal environment in order to determine the parameters of interfering signals within a pre-specified accuracy, and

- Expert system rules to select from a library of preselected techniques, suitable interference rejection schemes based upon the knowledge obtained from monitoring the signal environment.

The effectiveness of this novel interference rejection capability is demonstrated by computer-aided modeling and simulation using the software package $SPW^{©}$, a time-domain Signal Processing Worksystem.

Contents

List of Figures

Chapter 1

Introduction

Spread-spectrum signals are used widely in military and commercial communication systems due to their interference rejection capability and their lower probability of interception. In military applications the effects of intentional interference (jamming) are mitigated by the processing gain of the spread-spectrum system. In commercial systems code division multiple access (CDMA) systems rely heavily on this interference rejection ability to operate satisfactorily in the presence of multiuser interference.

Interference rejection in spread spectrum systems is achieved by pseudo-randomly distributing the information to be transmitted over a range of parameters (time, frequency and phase). This gives rise to what is known as the processing gain of the system. In many spread-spectrum systems, processing gain alone is not sufficient to achieve satisfactory system performance and additional interference rejection techniques need to be employed. The interference suppression circuit is placed prior to the spectrum despreader with the goal of reducing the jammer/interferer energy to an adequately low level that can be handled by the system processing gain.

The problem of interference rejection has been investigated quite extensively. A variety of interference suppression techniques have been presented in the literature [1, 2, 3, 4, 5]. These techniques can be broadly categorized into two classes:

- Time-domain adaptive filter cancellation
- Transform domain excisors.

While these schemes are able to provide effective cancellation for specific types of interference, no single scheme is able to suppress all types of interference that is encountered by the spread-spectrum system. We present a novel knowledge-based interference cancellation scheme for direct-sequence spread-spectrum systems. This innovative approach utilizes :

- *IPUS*, an expert system for the Integrated Processing and Understanding of Signals, to monitor the communication signal environment in order to determine the parameters of interfering signals within a pre-specified accuracy, and
- Expert system rules to select from a library of preselected techniques, suitable interference rejection schemes based upon the knowledge obtained from monitoring the signal environment.

The effectiveness of this novel interference rejection capability is demonstrated by considering a number of interference scenarios and using the software package $SPW^{©}$, a time-domain Signal Processing Worksystem.

In Chapter 2 we provide a brief tutorial on spread-spectrum signals and jammers. A brief review of existing interference cancellation algorithms is presented in Chapter 3. The overall framework of our knowledge-based cancellation approach is introduced in Chapter 4. Details of *IPUS* and how it isolates different interference signals are given in Chapter 5. Some architectural issues related to prototype development are discussed in Chapter 6. *SPW* models used in our system development are given in Chapter 7. Simulation results for three interference scenarios are presented in Chapter 8. Concluding remarks as well as recommendations for future work are given in Chapter 9.

Chapter 2

Spread-Spectrum Communication Systems

In all communication systems the bandwidth occupied by the modulated waveform is dependent upon the modulation method used and the data that is being transmitted. In a spread-spectrum system, the transmitted signal has a bandwidth that is much larger than the original message waveform bandwidth. The spreading of the message bandwidth over a wide frequency band limits the ability of an interfering signal to cause a significant distortion to the transmitted information. The flat noise-like spectrum of the spread-spectrum signal also hinders unwarranted interception of the signal. The use of a transmission bandwidth which is much larger than that required by the message data rate is responsible for the improved interference rejection capability. The following are three salient features of a spread-spectrum communication system [1, 2]:

- The bandwidth of the spread-spectrum signal is much larger than the bandwidth of the original information-bearing signal.
- Spreading is accomplished through the use of a spreading code that is independent of the data itself.

- Reception is accomplished by cross correlation of the received wide-band signal with a signal employing a synchronously generated replica of the spreading code.

Some of the popular techniques employed in spread spectrum communication systems are [2, 3]:

- Direct sequence
- Frequency hopping
- Time hopping
- Chirp
- Hybrid methods.

In this research effort we have concentrated on interference cancellation schemes for Direct-Sequence Spread-Spectrum (DSSS) systems.

2.1 Direct-Sequence Spread-Spectrum Systems

In direct-sequence spread-spectrum systems a pseudo-random noise (PN) code is used for spectrum spreading. The code sequence has a much higher rate than the original digital data rate which greatly expands the bandwidth beyond that of the original information bandwidth. This PN code, $c(t)$, takes values $+1$ and -1 and is known both to the transmitter and its dedicated receiver. As a result of multiplication by the PN code $c(t)$, the bandwidth of the information signal $d(t)$ is spread over a wider frequency band. The spread signal $s(t) = d(t)c(t)$ is transmitted after modulation and received in the presence of white noise and interference. In Figure 2.1, a base-band spread-spectrum communication system is shown. The base-band received signal is given by

$$r(t) = d(t)c(t) + n(t) + j(t) \tag{2.1}$$

where $n(t)$ and $j(t)$ denote the equivalent base-band white noise and interference, respectively. The received signal is despread at the receiver by multiplying with a locally generated replica of $c(t)$. After multiplication by $c(t)$, the resulting signal $y(t)$ can be written as follows:

$$\begin{aligned} y(t) &= c^2(t)d(t) + c(t)(n(t) + j(t)) \\ &= d(t) + c(t)n(t) + c(t)j(t). \end{aligned} \tag{2.2}$$

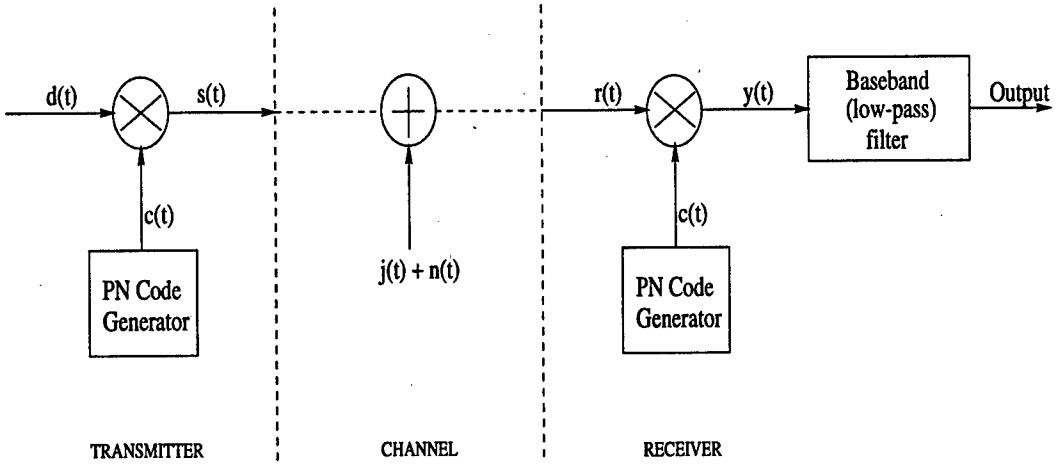


Figure 2.1: Base-band Spread-Spectrum System

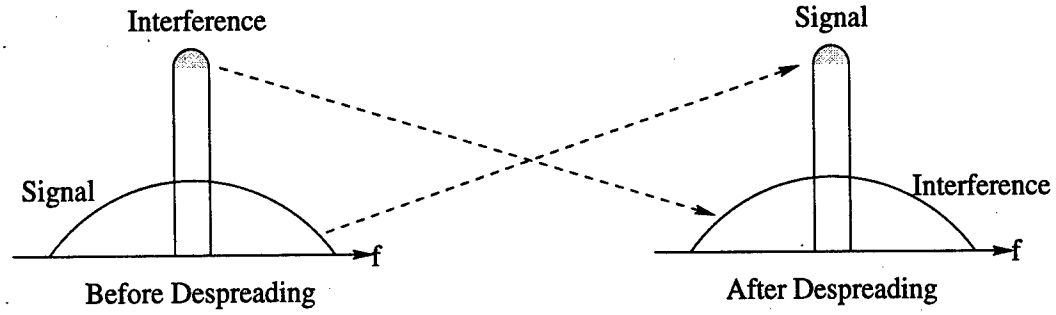


Figure 2.2: Effect of Spreading and Despreading on Signal and Interference Spectra

As shown in Figure 2.2, the desired signal component of $y(t)$ has a bandwidth that has been reduced back to the original data bandwidth and the interference bandwidth is increased to at least the spread bandwidth. When $y(t)$ is passed through a low-pass filter with bandwidth equal to that of $d(t)$, most of the noise and interference energy in $(n(t) + j(t))c(t)$ is removed by filtering. This gives rise to what is referred to as the processing gain advantage over the interference. The effective jammer power is reduced at the output of the low-pass filter, since all jammer components outside of the data bandwidth are rejected.

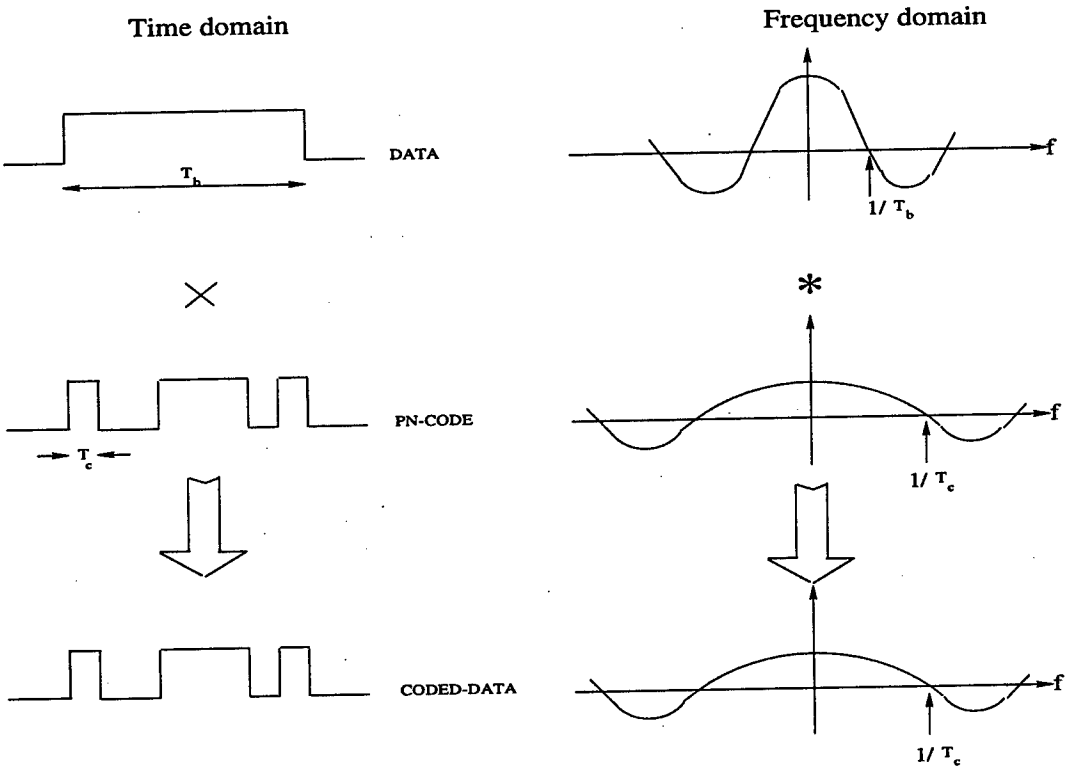


Figure 2.3: Spread Spectrum Signaling

As shown in Figure 2.3, the square pulse with duration T_b represents part of the binary information signal. Its Fourier transform is a Sinc function with zero values spaced at $1/T_b$. The spread-spectrum signal is formed by multiplying the information signal with a PN sequence consisting of narrow pulses of time duration T_c. Multiplication in the time domain is equivalent to convolution of the spectra in the frequency domain. Because the

bandwidth of the PN sequence spectrum is $1/T_c$, the spread-spectrum signal has a much larger bandwidth than that of the transmitted message whenever $T_c \ll T_b$. The basic pulse width in the PN sequence is T_c and is referred to as the chip duration. We define the multiplicity factor, or the processing gain of the system as $G = T_b/T_c$. Pseudo-noise codes are periodic in that the produced sequence repeats itself after a certain period of time. A maximal length linear feedback shift register PN code generator [2] is shown in Figure 2.4 where C_k equals either 0 or 1 for k=1, 2, 3, ...,n.

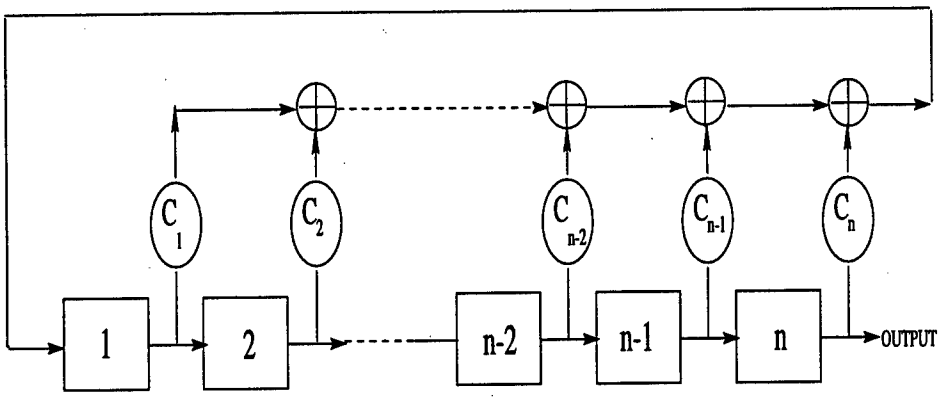


Figure 2.4: Maximal Length Linear Feedback Shift Register PN Code Generator

The PN code used in a spread-spectrum system should be designed such that the PN code is statistically independent of the message signal. In the case of a maximal length linear feedback shift register PN code generator, the period L [2] is equal to $2^n - 1$, where n is the number of stages in the code generator. An important reason for using maximal length PN sequences to modulate a message signal is the small correlation between successive chips as demonstrated by the autocorrelation function shown in Figure 2.5. It takes the value unity when the displacement $\tau = 0, \mp LT_c, \mp 2LT_c, ..$, and is equal to $-1/L$ [2] for all other values of τ. The period L of a PN sequence can be made large by employing a large number of stages in the shift register. In turn, the correlation between successive chips can be made as small as needed.

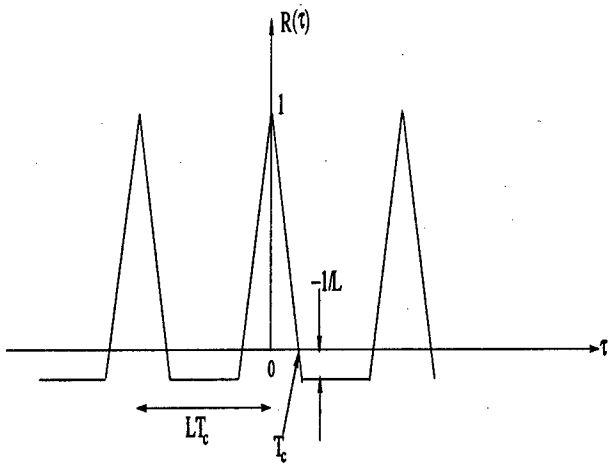


Figure 2.5: Autocorrelation Function of a Maximal Length PN Sequence

2.2 Jamming Signals

In this research we have focused our attention on military communication systems and intentional interferers (jammers). There are many types of jamming signals that could be considered. Different types of jammers affect the performance of DSSS systems differently. Some of the jammer types discussed below are considered in this study.

- CW and Multi-tone Jammers:

  Tone jammers transmit one or more narrow-band sinusoidal signals. These are spread across the desired bandwidth and are present all the time but do not occupy all of the frequency spectrum. When the tones are concentrated in only a portion of the spread spectrum signal bandwidth, the situation is frequently referred to as partial band jamming. A single tone CW jammer with average power J has the form

$$j(t) = \sqrt{2J}\cos(\omega t + \theta) \tag{2.3}$$

while multi-tone jammers using N_t equal power tones can be described by

$$j(t) = \sum_{l=1}^{N_t} (\sqrt{\frac{2J}{N_t}}) \cos(\omega_l t + \theta_l) \tag{2.4}$$

where all phases are assumed to be independent and uniformly distributed over $[0, 2\pi]$ and J denotes the average power of $j(t)$. The performance of direct-sequence spread-spectrum systems in the presence of a single-tone or multi-tone CW jammer is relatively easy to obtain. For example, the average probability of error for the multi-tone CW jammer case using BPSK is approximately given by [1, 3];

$$P_e = \frac{1}{2} erfc(\sqrt{\frac{E_b}{N_o + \sum_{n=1}^{m} J_n T_c}}) \tag{2.5}$$

where E_b is the bit energy, N_o is the white noise power spectral density, T_c is the chip duration, m is the number of jammer tones, J_n is the average power of the n^{th} jamming signal and erfc is the complement error function and given by

$$erfc(x) = \frac{2}{\sqrt{\pi}} \int_0^x exp(-z^2) dz.$$

- Pulse (ON/OFF) Jammer :

  A pulse jammer with average power J refers to a jammer which transmits a peak power given by

$$J_{peak} = \frac{J}{\rho} \tag{2.6}$$

for a fraction of time, ρ, and nothing for the remaining fraction of the time, $1 - \rho$. The received signal can be viewed as consisting of jammer plus thermal noise with probability, ρ, and thermal noise alone with probability, $1 - \rho$.

A direct-sequence spread-spectrum system that does not employ error-correction coding may be quite vulnerable to pulses that are longer than a message bit. The pulse jammer transmits short, high-intensity pulses, at either a regular or an irregular rate. A large duration pulse will completely destroy the message information contained in a given bit and, if these pulses occur with sufficient frequency, the resulting bit-error rate will be greater than an acceptable level. The net result of strong enough pulse jamming is that the probability of error increases and errors tend to cluster together. The jamming pulses are usually several message bits in length. The probability of error due to strong enough pulse jamming is given by

$$P_e \simeq \frac{1}{2}\rho \tag{2.7}$$

where ρ is the pulse duty factor. A duty cycle of 50% is frequently used, because it represents a suitable compromise between peak power, average power, and probability of error from the jammer's point of view.

- Narrow-band Interference:

  Interference signals whose spectral bandwidth is much smaller than the spectral bandwidth of the transmitted spread spectrum signal are classified as narrow-band interference signals.

- Linear-FM (Swept) Jammers:

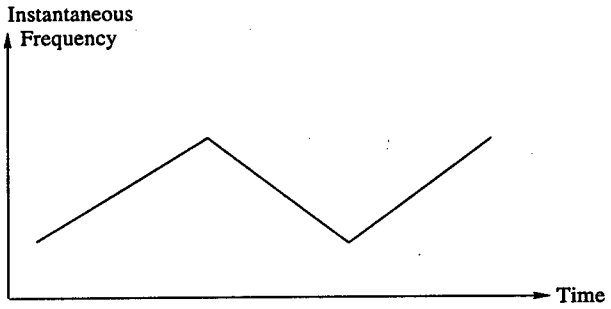


Figure 2.6: Instantaneous Frequency of a LFM Jamming Signal

This type of jammer is very effective in disturbing a direct-sequence spread-spectrum receiver. The instantaneous frequency of the jamming signal varies in a linear fashion with time. Depending on the rate of frequency change, these jammers could be classified further as fast linear-FM or slow linear-FM jammers. The range over which the jammer frequency is swept may include the entire spread-spectrum bandwidth or only a portion of this range. The instantaneous frequency is time variant (as shown in Figure 2.6) and the jamming signal with average power J may be expressed as

$$j(t) = \sqrt{2J}\cos(\int_0^t \omega(\tau)d\tau + \theta). \tag{2.8}$$

where $\omega(t)$ denotes the instantaneous frequency of the jammer.

- Broad-band Noise Jammer:

  A broadband noise jammer spreads noise of average power J evenly over the total frequency range of the spread bandwidth, ω_{ss}. This results in the equivalent single-sided noise power spectral density given by

$$N_j = \frac{J}{\omega_{ss}}. \tag{2.9}$$

  The broadband noise jammer is a brute force jammer that does not exploit any knowledge of the characteristics of the communication system except its transmission bandwidth, ω_{ss}. The resulting bit error probability is the same as that with additive white Gaussian noise of one-sided power spectral density equal to N_j.

Figure 2.7 illustrates different types of jammers. In the next chapter we discuss some of the existing interference cancellation techniques for DSSS systems.

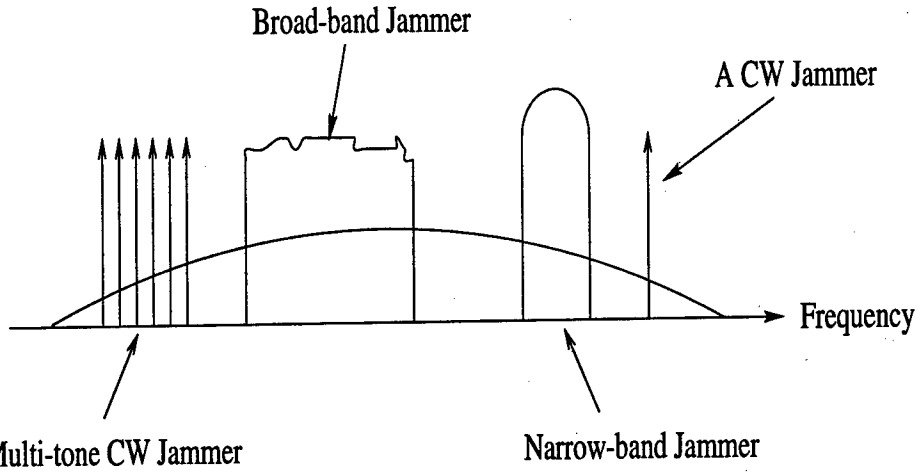


Figure 2.7: Frequency Domain Representation of Single and Multi-tone CW, Narrow-band, and Broad-band Jammers.

Chapter 3

Interference Rejection in Spread-Spectrum Systems

In a spread-spectrum communication system the effect of interference on system performance is reduced because of the inherent processing gain of the spread-spectrum system. To be able to achieve an even greater resistance to interference, the system processing gain (or spread-spectrum signal bandwidth) must be increased. In many practical applications this may not be possible due to bandwidth restrictions. Therefore, in many situations, especially in the presence of jammers, additional interference cancellation techniques need to be employed in order to attain satisfactory communication system performance. This problem has been investigated quite extensively. A variety of interference suppression techniques have been presented in the literature [4, 5]. These techniques can be broadly categorized into two classes:

- Time-domain adaptive filter cancellation
- Transform domain excisors.

The interference suppression circuit is placed prior to the spectrum despreader with the goal of reducing the jammer/interferer energy to an adequately low level that can

be handled by the system processing gain. These techniques introduce minor distortions in the received spread-spectrum signal. However, the performance improvement obtained by attenuating the interfering signals far outweighs such minor distortions caused by the canceller.

Interference rejection schemes for spread-spectrum systems often need to be adaptive because of the dynamic or changing nature of the interference and the channel. The design of optimum cancellation schemes requires a priori information about the statistical characteristics of the interference. We also need knowledge regarding the 'ON' times of the interference during which it is affecting communication performance. Perfect interference cancellation could be accomplished with the availability of accurate time-frequency characteristics of the interference. Adaptive cancellation techniques are used since such complete and accurate knowledge is generally not available. An adaptive technique is self-adjusting in nature and relies on a recursive algorithm to converge to a cancellation process that is optimum in some statistical sense. In the following sections, we discuss some of the existing interference cancellation techniques for DSSS systems.

3.1 Adaptive Transversal Filtering

There are many ways to implement adaptive filters to achieve narrow-band interference suppression. The most popular one is to use an estimator/subtracter method. The basic operation of this type of implementation involves subtracting an estimate of the interference from the received signal. This results in the received signal being whitened. An important problem faced in this approach is formation of the estimate. This is accomplished by employing the predictability of the interference. The spread-spectrum signal has an almost flat spectrum. Therefore, the spread-spectrum signal is unpredictable without the knowledge of the pseudo-random sequence used in spreading the spectrum. Since the interferer is narrow-band, a large portion of its energy is concentrated within a small bandwidth. Therefore, successive samples of the narrow-band interferer are strongly correlated and can

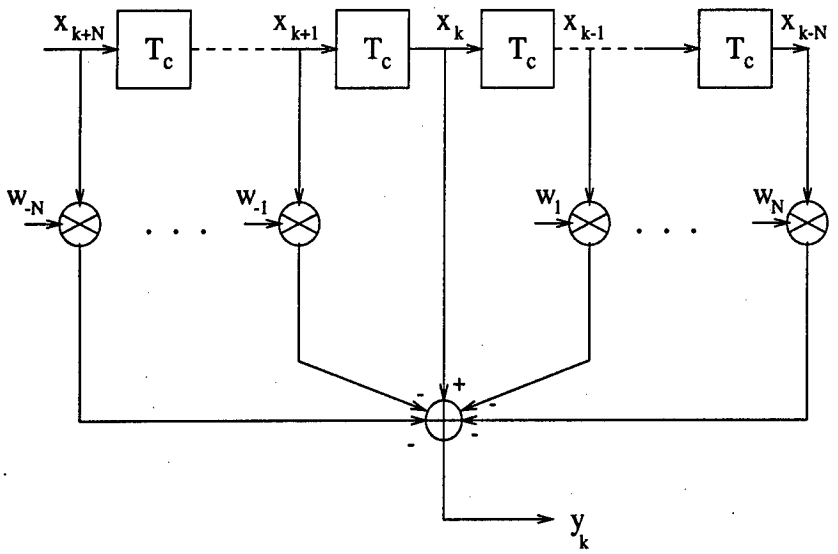
be closely predicted. Effective interference cancellation is then achieved by predicting the interference from the received signal and subtracting the predicted value from the received data samples. A transversal filter is often used as a predictor of the narrow-band interference signals. This adaptive transversal filter could have either a one-sided or two-sided filter structure as shown Figure 3.1, where T_c is the sampling interval which equals the chip duration.

If narrow band interference is present in the received signal, the weights of the transversal filter should be updated in order to predict the narrow-band interference in such a way that the resulting mean-squared error is minimized. Minimization can be done by using an algorithm such as the least-mean-squared (LMS) estimation algorithm which can be expressed as follows [4, 5]:

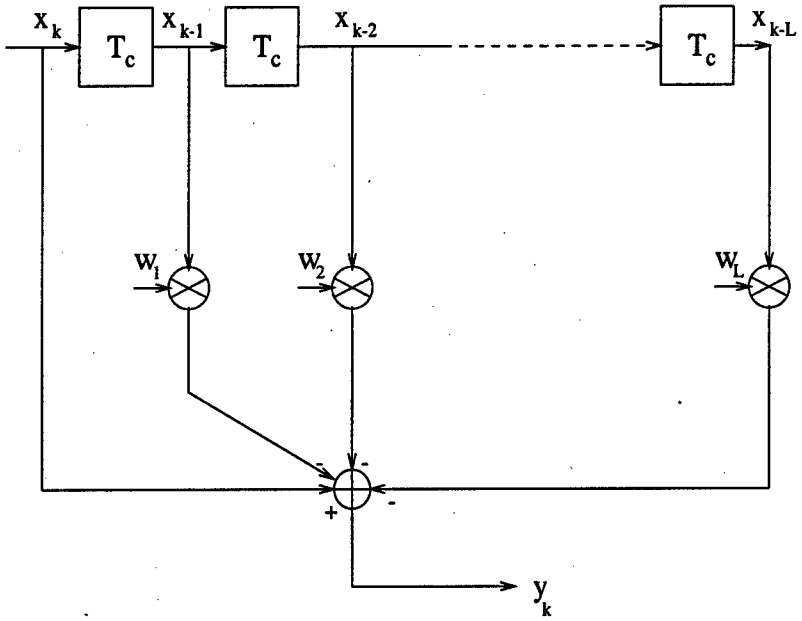
- 1) The filter output: $y_k = \underline{w_k}^H \underline{x_k}$
- 2) the error : $\epsilon_k = d_k - y_k$
- 3) The tap weight adaptation: $\underline{w_{k+1}} = \underline{w_k} + \mu \underline{x_k} \epsilon_k^*$

where k denotes the discrete time index, y_k is the scalar output of the transversal filter, $\underline{w_k}$ is the tap-weight vector, $\underline{x_k}$ is the tap-input vector, H indicates Hermitian transposition, ϵ_k is the estimation error, d_k is the desired response, μ is the step-size parameter and the asterisk denotes conjugation. In practice, several different design criteria are used to set the tap weights for the filter, as we discuss below.

Criterion 1 corresponds to whitening the received signal. It is based on predicting the narrow band interference and subtracting the estimate from the received signal. Since the transmitted spread spectrum signal and the white noise are wide-band processes, there is little correlation between their sample values. For this reason, it is impossible to predict future values of white noise and the transmitted spread-spectrum signal. However, there is always correlation between sample values of any narrow-band process. Hence, it is possible to predict with acceptable errors future values of the narrow-band interference from



(a)



(b)

Figure 3.1: (a): Two-sided Transversal Filter (b): Single-Sided Transversal Filter

past values. After having predicted the narrow-band interference signal, the transmitted spread-spectrum signal can be recovered by subtracting the predicted narrow-band interference from the received signal. Criterion 2 corresponds to whitening only the noise and interference. Criterion 3 corresponds to designing the filter so that an infinitely deep notch is placed at the instantaneous frequency of the narrow-band interfence. The level of the interference is reduced at the expense of introducing some distortion in the desired signal. When the interference is a single-tone jamming signal, increasing the power of this tone has no effect on system performance if the filter zero of the notch is accurately placed at the tone frequency. On the other hand, when the interference has a finite spectral width, it is impossible to reject the interference completely using a suppression filter with a finite number of taps since only a finite number of zeros can be placed in the frequency band spanned by the narrow-band interference.

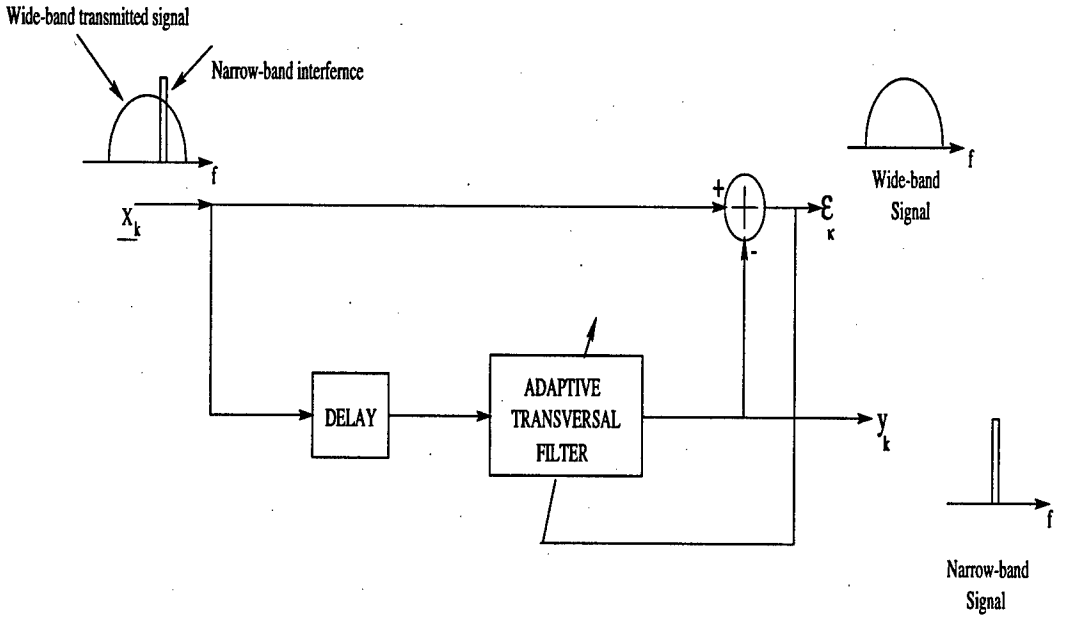


Figure 3.2: The Adaptive Transversal Filter Used as a Notch Filter or Whitening Filter.

It has been shown that the suppression filter is less effective against a source of interference with a finite spectral width than it is against a tone of equivalent power. It has also been shown that the performance against the narrow-band Gaussian process can be improved by increasing the number of taps in the transversal filter. In Figure 3.2 a block

diagram is given of the adaptive transversal filter used as either a notch filter or a whitening filter.

3.2 Transform Domain Processing

Transform domain processing [5, 6] can also be used to reject undesired signals and improve performance. The main idea is to pick a transform such that the interference will be approximately an impulse function in the transform domain while the transform of the desired signal is flat. One of the transforms which can satisfy the conditions mentioned above is the Fourier transform.

This cancellation scheme carries out the notch filtering operation in a different manner since cancellation is done in the frequency domain. In this processing, an electronic device which performs a real-time Fourier transform is used. This device is typically a surface acoustic wave (SAW) [7] device for spread-spectrum applications. Fourier transform based cancellation requires [4, 5] that

- The received signal be converted into the frequency domain by taking a real-time Fourier transform
- The sensed spectral peaks be suppressed through the use of a clipper with an adaptive threshold power level
- The signal be converted back to the time-domain by taking its real-time inverse Fourier transform.

Figure 3.3 illustrates the implementation of an adaptive transform domain receiver. Note that the envelope detector in the lower branch envelope detects the Fourier transform of the received signal and spectral peaks are fed into a clipper controlled by a threshold device. The upper branch passes the Fourier transform of the input directly to the multiplier. The clipper in the lower branch is set so that any time the output of the envelope detector

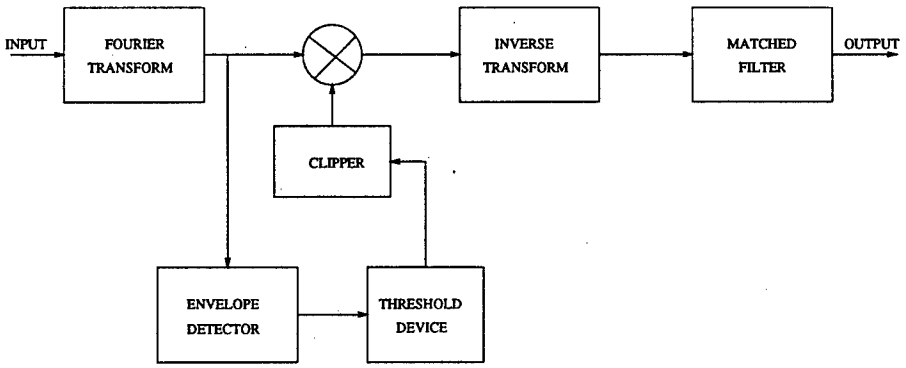


Figure 3.3: Block Diagram of an Adaptive Transform Domain Processing Receiver

exceeds a predetermined level, the output of the clipper is forced to some suitably small value. When this value is zero, we speak of infinite clipping. This operation is equivalent to an adaptive notching operation.

Transform domain techniques use methods, such as adaptive filtering, that are similar to time domain methods. However, the excision is done in the frequency domain. Although time-domain cancellation techniques, using adaptive transversal filtering, and frequency-domain cancellation techniques, using transform domain processing, accomplish the same result, the performance of each is drastically different. The adaptive time-domain cancellation techniques work extremely well provided the interference bandwidth is very small. However, they perform poorly when the interference bandwidth exceeds around five percent of the spreading bandwidth. In this case transform domain techniques work well.

3.3 Nonlinear Cancellation Techniques

The performance of an interference suppression filter can be evaluated in terms of its improvement in signal to noise ratio (SNR). For the purpose of such analyses, the base-band direct-sequence spread-spectrum signal can be modeled as an independent, identically distributed binary sequence. Note that such a sequence is highly non-Gaussian. Therefore, an optimum filter for predicting a narrow-band process in the presence of such a sequence will, in general, be nonlinear. The parameters can be estimated by an adaptive nonlinear

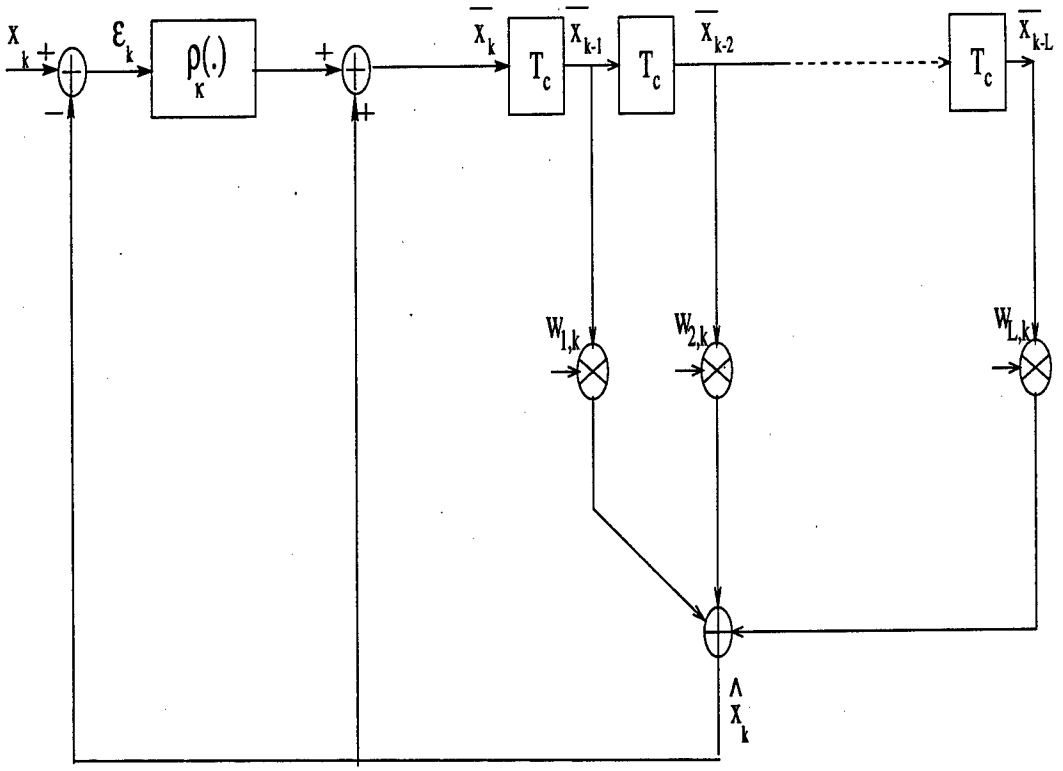


Figure 3.4: Nonlinear Adaptive predictor

filter that uses a standard LMS adaptation algorithm to predict the interferer by incorporating a nonlinearity that takes the form of a soft-decision feedback to estimate the spread-spectrum signal. As in the case of transversal filtering , the narrow band prediction is subtracted from the observation leaving the transmitted spread-spectrum signal plus white noise. An example of a non-linear adaptive predictor [8] is given in Figure 3.4 where the nonlinearity involves the tanh function. In particular, $\tanh(\frac{\epsilon_k}{\sigma_k^2})$ yields a soft decision on the spread-spectrum signal.

The LMS algorithm employed in this filter is given by [8]

$$\hat{x}_k = \underline{w_k}^H \underline{\bar{x}_k} \tag{3.1}$$

$$\epsilon_k = x_k - \hat{x}_k. \tag{3.2}$$

$$\rho_k(\epsilon_k) = \epsilon_k - tanh(\frac{\epsilon_k}{\sigma_k^2}) \tag{3.3}$$

$$\begin{aligned} \bar{x}_k &= x_k - tanh(\frac{\epsilon_k}{\sigma_k{}^2}) \\ &= \hat{x}_k + \rho_k(\epsilon_k). \end{aligned} \tag{3.4}$$

where x_k is the input signal, $\rho_k(\epsilon_k)$ is an estimate of the noise, $\hat{x}_k$ is an estimate of the interference signal, $\bar{x_k}$ is the observation less the soft decision on the spread-spectrum signal, ϵ_k represents the observation less the interference estimate, T_c is the delay equal to the chip duration, and L is the number of taps used in the transversal filter.

It has been shown that linear prediction techniques are suboptimal for prediction of a narrow-band interferer in the presence of non-Gaussian noise. Non-linear techniques have been shown to yield better performance.

3.4 CW Jammer Cancellation Using Phase-Locked-loop (PLL)

In this section we assume that the interference corrupting the transmitted spread-spectrum signal consists only of a CW jammer. In such a situation a PLL-based approach can be utilized to perform interference cancellation. Some theoretical background for the use of a PLL-based interference cancellation approach is given. For a spread-spectrum system operating in a CW interference environment, the received signal can be written as

$$r_1(t) = S_{ss}(t) + A_i\cos(\omega_i t + \theta_i) \tag{3.5}$$

where $S_{ss}(t)$ is the received spread-spectrum signal with average power S_R and bandwidth ω_{ss} and $A_i\cos(\omega_i t + \theta_i)$ is the received CW interference with average power $I_R = \frac{A_i^2}{2}$.

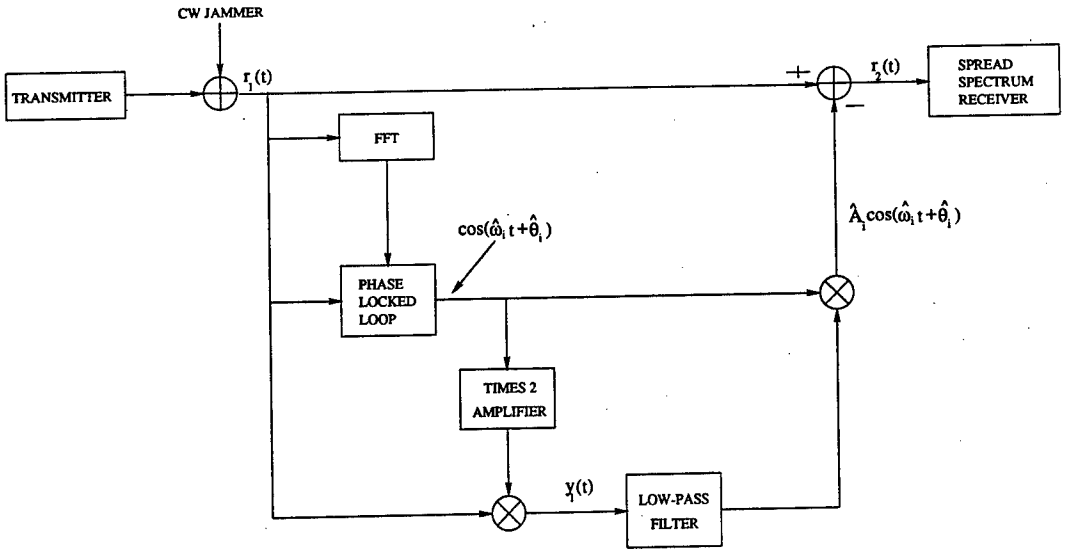


Figure 3.5: CW Interference Rejection System Based on a PLL

When the interference-to-signal ratio (ISR) is very high, the received signal consists of a relatively high interfering signal located at a single frequency plus a relatively low power spread-spectrum signal spread over a large frequency range. For this reason, an accurate estimate of the CW interfering signal can be obtained by using a CW interferer tracking circuit. This estimate can be used to reject the interfering signal prior to despreading and demodulation. One possible technique is to employ the PLL-based approach as shown in Figure 3.5. The net result is an input to the spread-spectrum receiver of

$$r_2(t) = S_{ss}(t) + A_i\cos(\omega_i t + \theta_i) - \hat{A}_i\cos(\hat{\omega}_i t + \hat{\theta}_i) \tag{3.6}$$

so that the effective interfering signal becomes

$$i_2(t) = A_i cos(\omega_i t + \theta_i) - \hat{A}_i cos(\hat{\omega}_i t + \hat{\theta}_i). \tag{3.7}$$

Assuming that the interference to signal ratio is high enough to ensure that the phase-locked loop is able to lock onto the CW interference such that $\hat{\omega}_i = \omega_i$, $i_2(t)$ is the difference

of two sinusoids of the same frequency which can be combined by the trigonometric identity

$$A\cos(\omega t + \alpha) + B\cos(\omega t + \beta) = C\cos(\omega t + \gamma) \tag{3.8}$$

where

$$C = \sqrt{A^2 + B^2 + 2AB\cos(\alpha - \beta)}. \tag{3.9}$$

Therefore,

$$i_2(t) = \sqrt{A_i^2 + \hat{A}_i^2 - 2A_i\hat{A}_i\cos(\theta_i - \hat{\theta}_i)}\cos(\omega_i t + \gamma). \tag{3.10}$$

The effectiveness of this interference rejection scheme is measured by the amplitude of $i_2(t)$ which is given by

$$A_{i_r} = \sqrt{A_i^2 + \hat{A}_i^2 - 2A_i\hat{A}_i\cos(\theta_i - \hat{\theta}_i)}. \tag{3.11}$$

Observe that the average power of $i_2(t)$ is

$$I_2 = \frac{A_{i_r}^2}{2}. \tag{3.12}$$

The amplitude estimate, $\hat{A}_i$, is generated at the output of the low pass filter (LPF) with input $y_1(t)$ where

$$y_1(t) = r_1(t)[2\cos(\omega_i t + \hat{\theta}_i)]. \tag{3.13}$$

Substituting for $r_1(t)$ from (3.5), (3.13) becomes

$$y_1(t) = 2S_{ss}(t)\cos(\omega_i t + \hat{\theta}_i) + A_i\cos(\phi_e) + A_i\cos(2\omega_i t + \theta_i + \hat{\theta}_i) \quad (3.14)$$

where $\phi_e = \theta_i - \hat{\theta}_i$. The response of the LPF to $y_1(t)$ provides the amplitude estimate of the CW interference and is

$$\hat{A}_i = A_i\cos(\phi_e) + E_a(t) \quad (3.15)$$

where $E_a(t)$ is the LPF response to $[2S_{ss}(t)cos(\omega_i t + \hat{\theta}_i)]$. With ϕ_e small, a good estimate of A_i can be obtained. $E_a(t)$ represents the noise in the amplitude estimate. The mean-squared value of $E_a(t)$ is given by

$$\overline{E_a^2(t)} = 2B_a(\frac{S_R}{\omega_{ss}}) \quad (3.16)$$

where B_a is the bandwidth of the LPF. Substituting (3.11) into (3.15) and assuming that $\phi_e << 1$, A_{i_r} is given by

$$A_{i_r} = \sqrt{A_i^2\phi_e^2 + E_a^2}. \quad (3.17)$$

The interfering signal power at the spread-spectrum receiver input is thus

$$I_2 = \frac{A_{i_r}^2}{2} \quad (3.18)$$

$$= \frac{A_i^2}{2}\phi_e^2 + \frac{E_a^2}{2} \quad (3.19)$$

$$= I_R\phi_e^2 + \frac{E_a^2}{2}. \tag{3.20}$$

For the case of interest here the PLL input is composed of $A_i\cos(\omega_i t + \theta_i)$ plus the wide-band spread-spectrum signal of average power S_R and bandwidth ω_{ss} which, to the narrow-band PLL, looks like white noise of spectral density S_R/ω_{ss}. Therefore, the phase error variance is given by

$$\sigma_{\phi_e}^2 = \frac{2S_R B_\phi}{A_i^2\omega_{ss}} \tag{3.21}$$

$$= \frac{B_\phi}{\omega_{ss}}\frac{S_R}{I_R} \tag{3.22}$$

where B_ϕ is the bandwidth of the PLL. If the phase estimate is unbiased, ${\phi_e}^2$ can be approximated by $\sigma_{\phi_e}^2$. Thus,

$${\phi_e}^2 = \sigma_{\phi_e}^2 \tag{3.23}$$

$$= \frac{B_\phi}{\omega_{ss}}\frac{S_R}{I_R}. \tag{3.24}$$

By using (3.24), (3.15), and (3.16) the interference-to-signal ratio, ISR, at the spread-spectrum receiver input can be expressed as

$$ISR = \frac{I_2}{S_R} = \frac{B_\phi + B_a}{\omega_{ss}}.$$

We conclude that when there is a single-tone jammer with a large average power which is contaminating the transmitted spread-spectrum signal, the PLL can achieve phase-lock and can generate an accurate estimate of the CW interferer. After cancellation, the residual interference due to incomplete interference rejection can readily be handled by the

processing gain (PG) provided by the spread-spectrum system.

3.5 Limitations of Existing Interference Cancellation Techniques

Interference cancellation schemes using adaptive transversal filters suffer from the drawback that they do not perform very well in the presence of multiple narrow-band interference signals and in situations where the instantaneous frequency of the interference varies rapidly over a large frequency range. Although these drawbacks may be partially overcome by increasing the order of the filter employed, this imposes additional problems in the form of filter transient times and receiver design complexity. Alternative approaches to narrow-band interference rejection involving time-frequency analysis performed using filter-banks have been proposed to handle scenarios with narrow-band interference with significant frequency modulation rates. While these solutions are suitable for suppressing a few interferers with slowly varying frequency modulations, they generally fail when applied to signals with several interferers with rapidly varying modulations. The primary cause for failure is the fixed tradeoff between time and frequency resolution at any given point in the time-frequency plane. When the interference has a finite spectral width, it cannot be completely rejected by suppression filters with a finite number of taps, since only a finite number of zeros can be placed in the frequency band spanned by the interference.

In transfer domain processing the primary consideration in the overall system design is the shape of the window used to view the received waveform. In the Fast Fourier transform (FFT) operation, rectangular windows are the ones most widely used. But it is also known that rectangular windows create very large side-lobes that severely degrade the interference cancellation performance [6]. Although these side-lobes can be reduced by proper weighting functions [6], they distort the input signal itself.

Nonlinear cancellation techniques have the same drawbacks as adaptive cancellation

techniques. In addition to this, it usually involves a very complex design. The main idea in nonlinear cancellation techniques is to optimize the detection process dynamically, in the presence of interference, by estimating the non-Gaussian statistics and using this information to derive a nonlinear transformation to apply to the corrupted signal. That is the reason why these cancellation schemes are quite complex.

In the next chapter we introduce our novel knowledge-based interference cancellation framework that is quite robust and is able to remove different types of interference signals.

Chapter 4

Knowledge-Based Interference Rejection Framework

As indicated in the previous chapter, a great deal of effort has gone into the development of interference cancellation schemes for spread-spectrum systems. These schemes are able to handle specific classes of interference. There is no single interference cancellation scheme that is able to effectively suppress all types of interference. Accurate knowledge of the time-frequency characteristics of the interference embedded in the received spread-spectrum signal is required in order to apply suitable cancellation techniques. In this chapter we introduce our innovative knowledge-based approach for interference cancellation in spread-spectrum systems. This approach utilizes

- *IPUS*, an expert system for the Integrated Processing and Understanding of Signals, to monitor the signal environment in order to correctly determine the interfering signals within a pre-specified accuracy

and

- Expert system rules to select from a library of preselected techniques suitable interference rejection schemes based upon knowledge obtained from monitoring the environment.

The approach has been found to be quite effective. Its effectiveness is demonstrated by computer-aided modeling and simulation using the software package SPW later in this report.

The overall knowledge-based framework for interference rejection is shown in Figure 4.1. The transmitted spread-spectrum signal is assumed to be corrupted by additive thermal noise and a number of interference signals prior to its reception at the receiver. The received signal is processed by the expert system called $IPUS$ which carries out the environmental monitoring function. It determines the types of interference signals (from a specified list of interference types) present in the received signal along with their parameters. If an interference type is not on the specified list, $IPUS$ can be extended to include new interference types. Details on the version of $IPUS$ included in our prototype will be provided in Chapter 5. Based on the information generated by $IPUS$ (time-frequency locations of the isolated interferers), the most appropriate interference cancellation filters from a library of available filters is employed to eliminate interference from the received signal. This selection is based on a set of rules that have been developed and described in Chapter 7. Additional rules can always be added when needed. Two switches are included to allow for either selection of an appropriate interference cancellation filter from the filter bank or for bypassing the interference cancellation stage when $IPUS$ determines that no interference is present. Finally, the spread-spectrum receiver demodulates its input and recovers the transmitted information.

The problem of interference cancellation in spread-spectrum systems may be formally described by considering received signals of the form:

$$r(t) = s(t) + \sum_{k=1}^{N} j_k(t). \tag{4.1}$$

where $s(t)$ is the information-bearing spread spectrum signal and $j_k(t)$, k=1,2,..., are different types of jamming signals that may be simultaneously present on the channel. They have the form $j_k(t) = a_k \cos[\int_0^t \omega_k(\tau)d\tau + \phi_k]$. Our objective is to sufficiently attenuate all the interferers $j_k(t)$ such that the original spread-spectrum signal $s(t)$ can be successfully demodulated due to the inherent processing gain of the system. In the prototype that we have developed to demonstrate the proof of concept for our knowledge-based approach, we have restricted the types of interferers to be:

(1) Multiple tone interferers operating in a continuous wave (CW) mode,

(2) Linear-FM (or chirp) interfering signals,

(3) Multiple CW tones operating in an ON/OFF mode with or without frequency hopping.

Different interference cancellation schemes are required to mitigate these different types of interferers. Also, for linear FM interferers, changes in the parameters of the cancellation filter are required as a function of time. For situations where one or more of these types of interferers are present, the $IPUS$-based interference cancellation approach is expected to show improved performance over that achieved with a single fixed interference canceller. For example, the instantaneous frequency information of linear FM interference available from $IPUS$ will suggest the use of a moving notch filter for interference cancellation.

Details on different aspects of our methodology are provided in the rest of the chapters. Chapter 5 describes the approach used in $IPUS$. In Chapter 6 we briefly describe the computer-aided design and simulation tool SPW. Integration of $IPUS$ and SPW is also discussed. Chapter 7 contains a description of communication system models used for simulation purposes. We also present some experimental work leading to the set of rules for interference cancellation filter selection. Several simulation examples to demonstrate the superior performance of our knowledge-based approach are presented in Chapter 8. Conclusions and recommendations for future work are provided in Chapter 9.

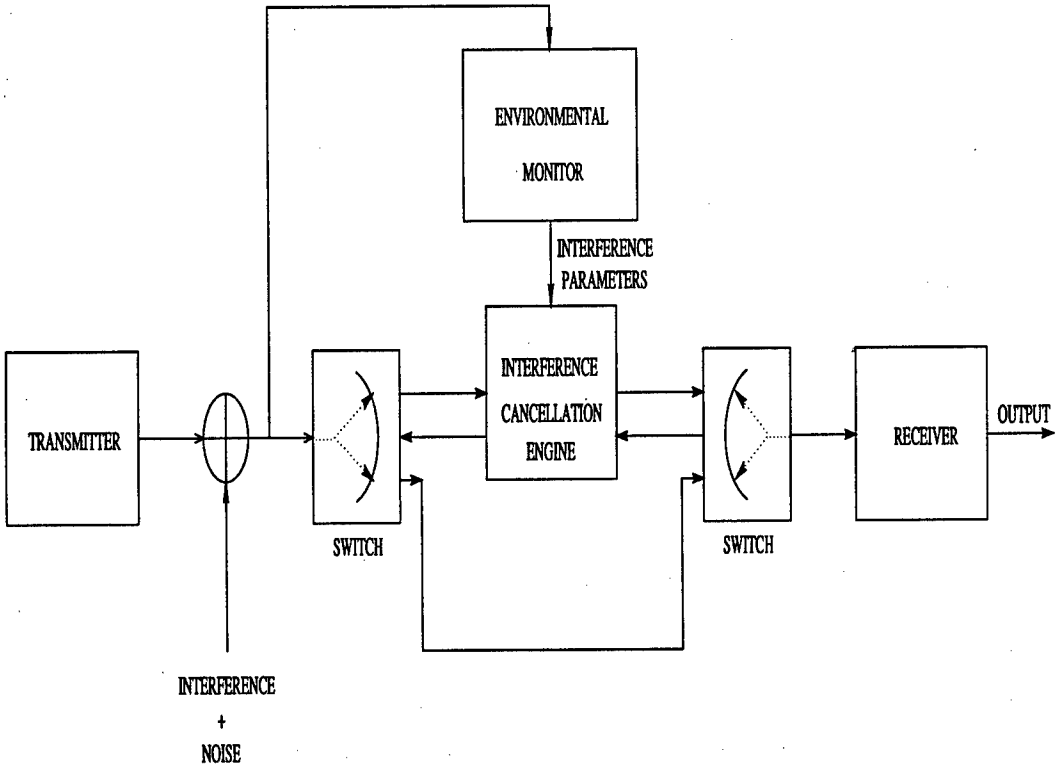


Figure 4.1: Block Diagram of the Knowledge-Based Interference Cancellation System.

Chapter 5

IPUS-Based Interference Isolation

In this chapter we provide a detailed treatment of our approach to isolating each modulated narrow-band interference (MNBI) tone from a contaminated spread-spectrum signal. In particular, we describe how we can efficiently search for a time-frequency (TF) representation of the signal that separately resolves the spectral contributions from each MNBI tone. Furthermore, we discuss the appropriateness of the *IPUS* framework [9] as the supporting system architecture for this type of search process. We also provide details on how we have incorporated our TF analysis approach into an *IPUS* framework. Finally, we describe a software system we have implemented and used to experimentally validate this approach to interference isolation.

5.1 Elaboration of Approach to Interference Isolation

Let us now consider the details of our approach for isolating MNBI tones from a contaminated spread spectrum signal. The process of such isolation may be alternatively viewed as the resolution of each MNBI tone from its nearest neighbor in a TF representation of the signal. Our overall strategy for accomplishing our isolation objective may be viewed as consisting of two main stages. In the first stage, we compute a TF representation using the

short-time Fourier transform (STFT) [10] in order to identify TF trajectories[1] that correspond to MNBI tones. The analysis filters of the STFT have sufficiently small bandwidths to ensure isolation of components that are as close as 16 Hz. For MNBI tones with significant frequency modulation (FM) such TF analysis leads to TF trajectories that are smooth versions of the original TF trajectories. During the second stage the TF trajectory of each MNBI tone with FM is refined by re-processing the signal using an adjusted filterbank. The filters in this filterbank are modified in a manner such that they optimize the time resolution of MNBI tones with FM while resolving them separately from their neighbors.

5.1.1 Delineating Isolated MNBI Tones

In the first stage we find time-frequency-amplitude trajectories for all MNBI tones present in the signal. To find such trajectories, we process the signal using an STFT filterbank whose analysis filters have impulse responses of the form

$$h_c(t) = h_0(t)e^{j\omega_c t} \tag{5.1}$$

where ω_c is the center frequency of the filter and $h_0(t)$ is a Gaussian waveform that controls the bandwidth of the filter. The bandwidth of each filter is assumed to be less than 32 Hz because we expect MNBI tones to be no closer than 16 Hz. The output of the STFT filterbank separately resolves each MNBI tone from its nearest neighbor. Spectral peak picking and tracking algorithms determine the time-frequency-amplitude trajectories of MNBI tones from the output of the STFT.

[1]For the purpose of this discussion, we assume the availability of algorithms that can detect and track peaks in the TF representation in order to form time-frequency-amplitude trajectories for the various tones present in the signal.

5.1.2 Adjusted Filterbank Analysis of MNBI Tones with FM

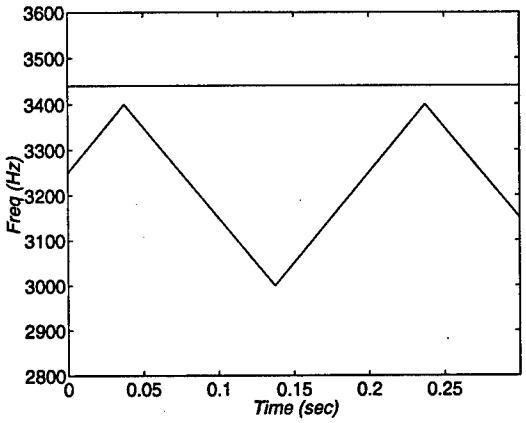
During the second stage we refine the TF trajectories of MNBI tones with FM. The narrow bandwidth filters utilized in the first stage provide inaccurate estimates for the TF trajectories of such tones. The process of refining these trajectories involves adjusting the analysis filters in the vicinity of each FM tone to have optimal time resolution *and* sufficient frequency resolution to separate the tone from its nearest neighbor.

To carry out such filtering in the vicinity of MNBI tones with FM we adopt an iterative procedure for adjusting the filters. For a candidate MNBI tone we first determine the frequency separation $\Delta f(t)$ from its nearest neighbor using the TF trajectories obtained in the first stage. Furthermore, we also use the time-amplitude trajectories of the MNBI tones found in the first stage to determine the relative amplitude $a_r(t)$ of the candidate tone with respect to its nearest neighbor. The time-varying functions $\Delta f(t)$ and $a_r(t)$ are utilized to determine the bandwidth of the analysis filters centered around the candidate tone for each time. This bandwidth is such that it allows only 2% of the candidate tone's energy to be contributed by the nearest neighbor. To obtain the filters with the desired bandwidth, the variance of the Gaussian $h_0(t)$ for the corresponding filter should be chosen to be

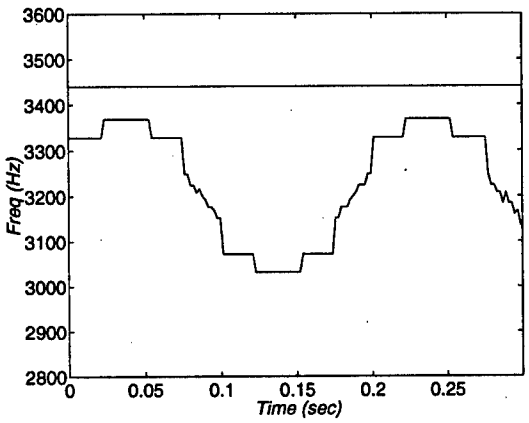
$$\sigma(t) = \frac{2.14\sqrt{a_r(t)}}{2\pi\Delta f(t)}. \tag{5.2}$$

These adjusted analysis filters are applied to the signal and a refined time-frequency-amplitude trajectory of the FM tone is formed. If the average percentage change between the previous TF trajectory and the refined TF trajectory is less than 5%, we assume that the filters have succeeded in refining the trajectory of the candidate tone. For larger changes, we repeat the process described above but with the refined estimate for the time-amplitude and time-frequency trajectories. At the end of the iterative process, an accurate TF trajectory of the candidate tone is obtained.

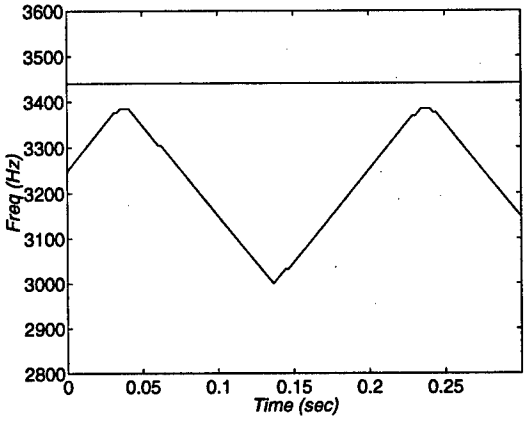
We illustrate the benefits of adjusting the analysis filters through an example. The signal scenario depicted in Figure 5.1(a) consists of two closely spaced tones: one with no



(a)



(b)



(c)

Figure 5.1: Illustrating the benefits of adjusting the analysis filters. (a) Signal scenario. (b) Results of using stage 1. (c) Results of using filters with bandwidths that are adjusted to take the FM tone into account.

modulation and having a frequency of 3440 Hz and the other with FM and centered around 3200 Hz. The TF trajectories from the first stage of processing are shown in Figure 5.1(b). We see that the trajectory of the FM tone is inaccurate. We now use the results of Figure 5.1(b) to estimate the frequency separation and relative amplitude of the two tones. Based upon these estimates, we obtain a modified set of analysis filters and re-process the signal. The results of this processing are shown in Figure 5.1(c). The TF trajectory of the FM tone is now similar to the original FM tone TF trajectory.

5.2 Supporting System Architecture

We now examine our reasons for selecting the *IPUS* framework as the supporting system architecture for our approach to interference isolation. Because of its blackboard architecture, the *IPUS* framework allows constraint matching and signal re-processing to be carried out on data at multiple levels of abstraction. This is suitable for a system realization of our approach which requires the application of constraints on TF trajectories of MNBI tones. Furthermore, the blackboard paradigm emphasizes modularity that is conducive to incremental system development. Finally, a distinguishing feature that sets the *IPUS* framework apart from other blackboard architectures is a central mechanism called the re-processing loop. This mechanism allows convenient system realization of our two-stage approach to TF analysis described in the previous section.

In the next sections, we discuss the basics of the *IPUS* model's re-processing loop and the description of how this re-processing loop supports various aspects of our two-stage approach is provided.

5.2.1 IPUS Model

Central to the *IPUS* model is a *reprocessing loop* which detects and eliminates inconsistencies between signal processing results and constraints on permissible signal behaviors.

The reprocessing loop is iterative because the elimination of inconsistencies generally requires a search over various plausible remedial actions. During each iteration processing is carried out on signal representations that are stored at various levels of abstraction in a blackboard database. Algorithms for constraint matching and signal reprocessing, which are called knowledge sources (KSs), operate on these data abstractions. The KSs utilized in the reprocessing loop are of four types: discrepancy detection, discrepancy diagnosis, reprocessing planning, and reprocessing. In Figure 5.2, we show how these KSs are invoked during the reprocessing loop.

The *discrepancy detection* KS identifies "distortions" in the signal features extracted from the results of signal processing. Such distortions correspond to features that are incompatible with constraints from the application domain. Alternatively, distortions also arise when signal features are inconsistent with expected signal features. A prediction KS is invoked prior to the reprocessing loop to generate the expected features on the basis of signal features found in earlier temporal intervals of the signal data.

Distortions identified during discrepancy detection are analyzed by a *discrepancy diagnosis* KS. This KS generates plausible explanations for the cause of each distortion. The generation of these explanations may be viewed as the process of using one or more "distortion operators" to map expected signal features to the signal features detected in the processing interval. We illustrate such mapping through an example. Consider a signal in the spread spectrum domain for which expected features indicate the presence of two tones, while signal features detected from signal processing outputs show the presence of a single tone. Diagnosis would then use a frequency resolution type of distortion operator to map the two tones to a single tone. To facilitate such mapping the diagnosis KS generally has an intimate knowledge of domain constraints as well as the properties of signal processing algorithms used in the application. It should also be noted that there may not be a unique sequence of distortion operators to explain a distortion. Consequently, the diagnosis KS is not guaranteed to produce the "correct" explanation in the first attempt. Multiple itera-

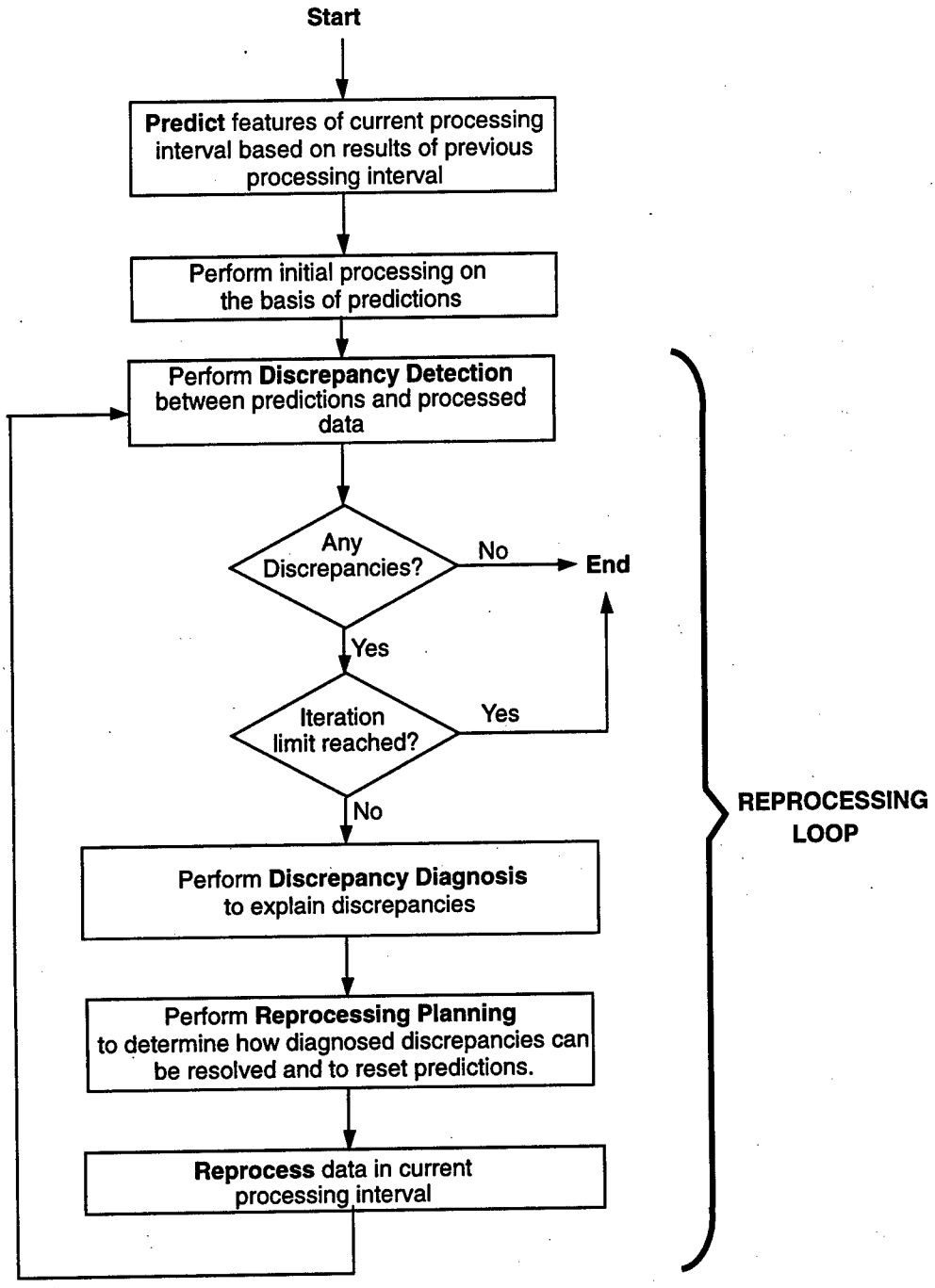


Figure 5.2: IPUS model.

tions of the reprocessing loop need to be performed to sift through the explanations and converge on the correct one.

Following diagnosis, a *reprocessing planning* KS utilizes the distortion operators to hypothesize remedial reprocessing KSs that could be applied to the data. This requires that the reprocessing planning KS have an accurate knowledge of the behaviors of various signal processing algorithms. We illustrate reprocessing planning using the example considered in the description of discrepancy diagnosis. In this example, a low frequency resolution distortion operator was identified to explain the distortion of two tones into a single tone. When the reprocessing planning KS finds such an operator, it hypothesizes that the signal should be reprocessed with filters having smaller bandwidths. The new set of filters are then designed on the basis of the relative amplitudes of the tones. In certain situations, reprocessing planning cannot arrive at the best reprocessing KS in the first iteration of the reprocessing loop. Multiple iterations of the reprocessing loop are again necessary to ensure that the correct strategy is identified.

Reprocessing KSs are finally applied to the signal data to try and remove the distortions. Once reprocessing has been performed, the reprocessing loop is repeated to identify and resolve distortions that may still persist. Persistant distortions result from incorrect diagnosis or the reprocessing planning failing to specify proper remedial actions. Repetitions of the loop are carried out until all distortions are removed or an iteration limit is reached.

5.2.2 Appropriateness of IPUS Model

We now examine how the various mechanisms of the $IPUS$ framework's reprocessing loop are utilized to obtain a system realization of our two-stage TF analysis approach. In Figure 5.3 we depict the specific mapping that we have performed between each of the two stages and the $IPUS$ mechanisms of prediction, discrepancy detection, reprocessing planning, and reprocessing. Let us examine this mapping in greater detail.

We begin by using the prediction and discrepancy detection mechanisms to obtain an

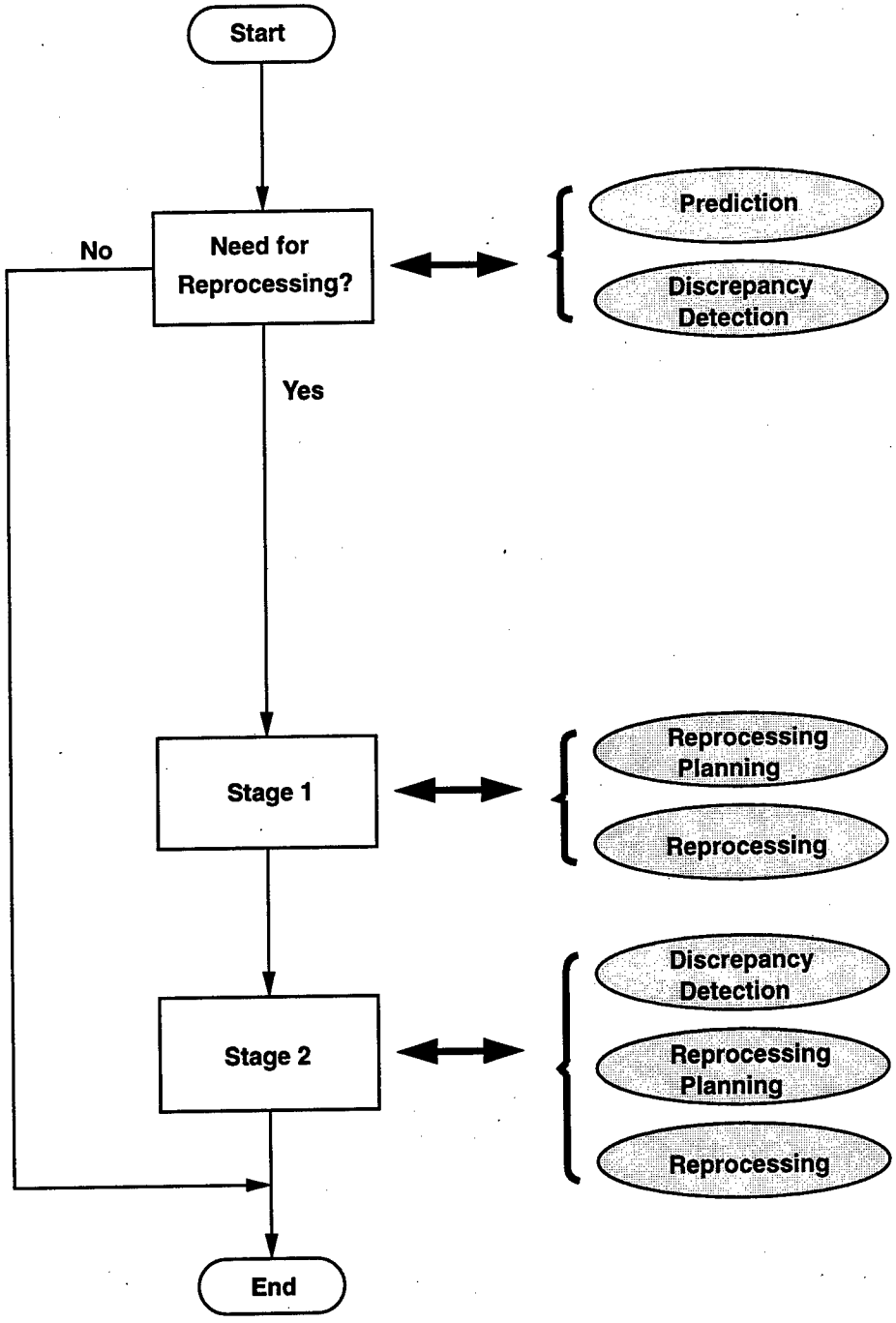


Figure 5.3: Depiction of the mapping between various aspects of our MNBI isolation approach and IPUS mechanisms.

initial TF representation of the signal and to identify subregions of this representation that need to be reprocessed. The prediction mechanism is utilized to hypothesize the TF trajectories of tones in the current signal interval based upon existing tone hypotheses from past signal intervals. This allows us to make a priori adjustments to an STFT filterbank and apply it to the current signal interval. TF trajectories obtained from the resulting initial TF representation are compared against the predictions by using the discrepancy detection mechanism. Such a comparison helps identify TF subregions where the initial TF representation needs to be reprocessed for ensuring that the tones in the signal are resolvable.

Reprocessing planning and reprocessing mechanisms are utilized to carry out the first stage of our TF analysis approach on discrepant TF subregions corresponding to interacting MNBI tones. The reprocessing planning mechanism helps in designing the analysis filters corresponding to each MNBI tone. During this design, it is assumed that no two MNBI tones are less than 16 Hz apart. Reprocessing is then carried out using the analysis filters and the time-frequency-amplitude trajectories of MNBI tones are extracted.

Finally, a combination of discrepancy detection, reprocessing planning, and reprocessing are used to carry out the second stage of processing in our approach. The discrepancy detection mechanism checks the TF trajectories obtained during reprocessing in the first stage to find any MNBI tones with FM. Discrepant MNBI tones are then targeted for reprocessing using adjusted analysis filters. The reprocessing planning mechanism helps in adjusting the bandwidth of the filters to obtain optimal time resolution for each MNBI tone while resolving the tone from its nearest neighbor. Reprocessing is carried out using the adjusted filters and the TF trajectories of tones are extracted from the filter outputs. These trajectories are again checked by the discrepancy detection mechanism to determine if they have been sufficiently refined. The combination of discrepancy detection, reprocessing planning, and reprocessing is iteratively applied until the analysis filters for each MNBI tone with FM provide an accurate estimate of the tone's TF trajectory. This completes

the *IPUS*-based TF analysis of the signal interval.

5.3 IPUS Mechanisms for TF Approach

In this section we describe the specific details on how the *IPUS* framework was used to support our approach to TF analysis. We begin with a description of the data representations used in the various mechanisms of the reprocessing loop. Later we describe the prediction, signal processing, discrepancy detection, reprocessing planning, and reprocessing KSs that we have developed.

5.3.1 Data Representations

For facilitating prediction, discrepancy detection, diagnosis, and reprocessing, we need to represent the signal data in terms of modulated tones. Such tone representations may be obtained through the application of spectral peak picking and tone tracking to a TF representation of the signal. TF analysis first separately resolves spectral contributions from each tone in the TF plane. Significant spectral peaks are identified in the filterbank outputs using Klassner's peak picker [11]. The peaks are then used to form time-frequency-amplitude trajectories of the modulated tones present in the signal.

The tracking of trajectories of modulated tones may be viewed as a problem of *multi-target tracking* [12]. A solution to this problem involves modeling the time-frequency-amplitude evolution of a tone by means of *state equations* and then utilizing *multi-hypothesis target tracking* [13] to form state trajectories for each modulated tone. The state equations that we utilize are based upon those introduced for maneuvering targets [14] and are given by

$$\begin{aligned} \dot{x}(t) &= Fx(t) + Gv(t) \\ z(t) &= Hx(t) + w(t) \end{aligned} \tag{5.3}$$

where $x(t)$ is expressed in terms of the modulated tones instantaneous frequency $f(t)$ and instantaneous amplitude $a(t)$ as

$$x(t) = \begin{bmatrix} f(t) \\ \dot{f}(t) \\ \ddot{f}(t) \\ a(t) \\ \dot{a}(t) \\ \ddot{a}(t) \end{bmatrix}, \quad F = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -\alpha & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -\alpha \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad H = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}^T . \quad (5.4)$$

$v(t)$ and $w(t)$ are Gaussian white noise processes, and $z(t)$ is the measurement vector corresponding to the frequency and amplitude of a peak. As outlined in Figure 5.4, the trajectory of the current state of the modulated tone is formed through a procedure involving peak association followed by Kalman filtering for updating the state. A set of hypotheses are formed at the current processing time through different associations of the current peaks with tracks. For every hypothesis, its probability is computed based upon the current state of each track and by making suitable assumptions about the probability of detection of a peak corresponding to an existing track. Finally, the hypothesis with the highest probability is chosen and the tracks are updated by using the Kalman filter. The tracking process is continued until all peaks have been assimilated into tracks.

5.3.2 Prediction

Prediction is the process of using previously identified tones in order to hypothesize their TF evolution in the current processing block. We perform prediction on a modulated tone by first carrying out a least-squares straight-line fit [15] to its corresponding track. The standard deviation of the tone's instantaneous frequency is also computed. The straight line is then extrapolated into the current processing block and a frequency subregion which is 3 standard deviations to either side of this line is identified. Assuming a normal distribution

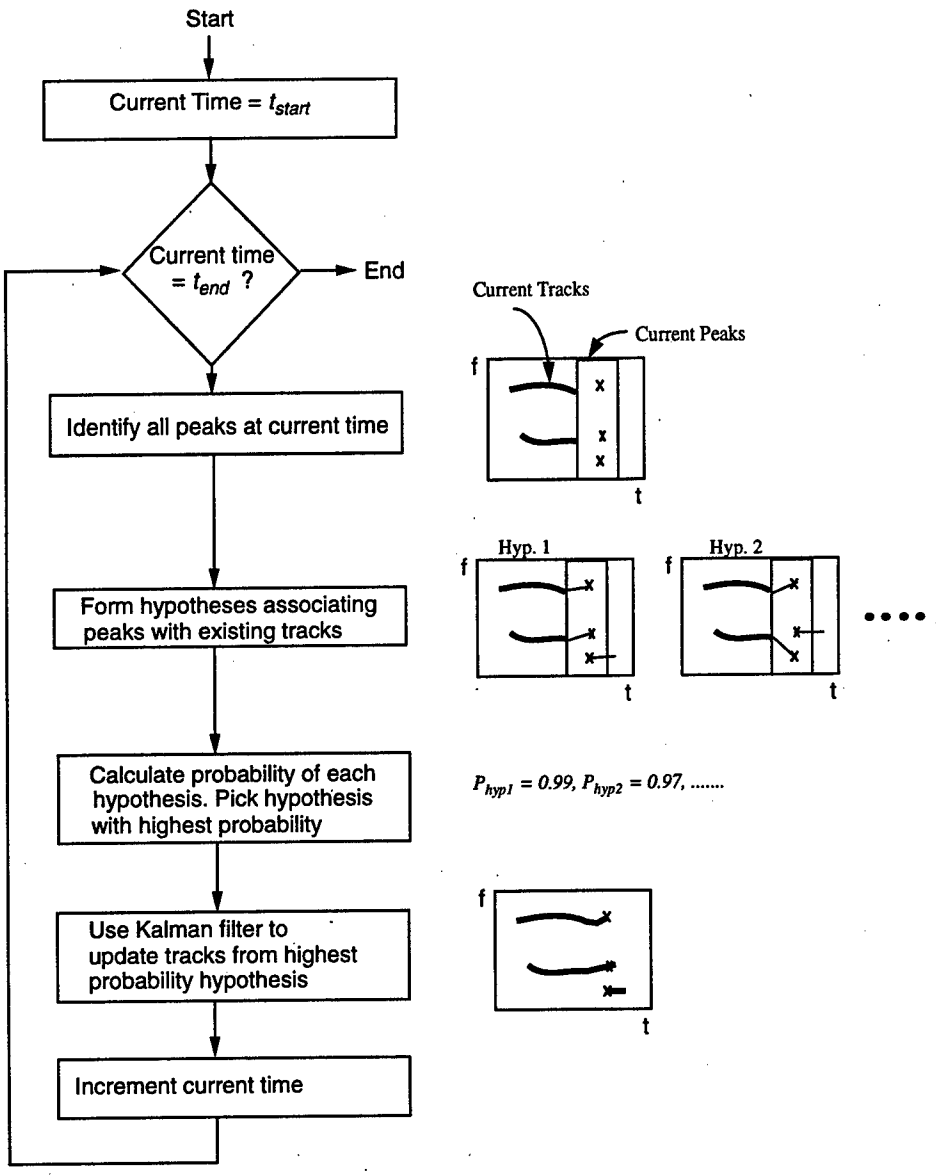


Figure 5.4: Procedure for tracking the time-frequency trajectories of tones on the basis of spectral peaks extracted from the signal.

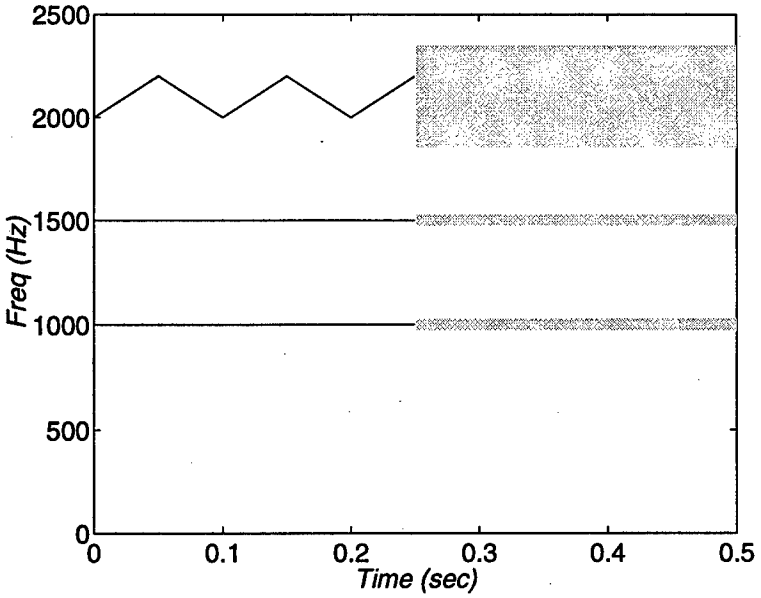


Figure 5.5: Illustrating the extrapolation of TF trajectories of tones from the previous processing block to the current processing block. The gray regions indicate the TF regions over which the tones identified in the previous block (indicated by black lines) are most likely to evolve.

for the tone's instantaneous frequency, this subregion represents the 99% confidence region for the evolution of the trajectory corresponding to the tone. For instance, in Figure 5.5, the gray regions indicate the TF regions over which the tones identified in the previous block (indicated by black lines) are most likely to evolve. These TF regions in conjunction with mean amplitude and frequency modulation rate extracted from the tracks corresponding to the tones constitute their overall predicted TF behavior.

5.3.3 Adjusted STFT Processing

The predicted TF behavior of tones in the current processing block is utilized to alter the analysis filters in an STFT filterbank with Gaussian filters. Gaussian filters were chosen on the basis of the well-known property that they have the least time-frequency uncertainty [16]. The center frequencies of the filters in the filterbank are uniformly spaced along the

frequency axis and the impulse response of the i-th filter is given by

$$h_i[n] = \begin{cases} A\exp\{-\alpha n^2\}\exp\left\{j\frac{2\pi f_i}{f_s}n\right\}, & |n| \leq n_i \\ 0, & |n| > n_i \end{cases} . \tag{5.5}$$

Here f_i is the center frequency of the filter, f_s is the sampling rate, n_i is the time index before which the magnitude of the impulse response decays to a small value ϵ, A is a scaling factor which ensures that the filter has unit energy, and α is a parameter that controls the bandwidth of the filter. To ensure that the filters have bandwidths with adequate frequency resolution to separately resolve two adjacent tones, we have to pick an appropriate value of α for filters in the vicinity of the two tones. Through an analysis of the Gaussian filter's frequency response, we have shown that this may be achieved by ensuring that the α parameter corresponding to these filters is at most

$$\alpha = \frac{8\pi^2}{\ln(\epsilon)}\frac{A_1^2|f_1 - f_2|^2}{(3A_2 - A_1)^2 f_s^2} \tag{5.6}$$

where A_1 and A_2 are the average amplitudes of the two predicted tones such that $A_2 > A_1$ and f_1 and f_2 are the frequencies of the two predicted tones.

5.3.4 Discrepancy Detection

Two different discrepancy detection mechanisms are specified for use before stage 1 and during stage 2 of our approach, respectively. The first mechanism is targeted at finding discrepancies between the predictions and the results of initial STFT processing. The second discrepancy detection mechanism is used in adjusting the analysis filters for MNBI tones with FM in stage 2.

Discrepancy Detection before Stage 1:

Discrepancy detection is the process of identifying mismatches between the predictions and the results of adjusted STFT processing. In order to facilitate discrepancy detec-

tion, the results of initial STFT processing are used to compute temporal and spectral energy profiles of the current analysis interval. The temporal profile corresponds to integrating the magnitude-squared of the output of the STFT over the entire time range of the current analysis interval. Similarly, the spectral energy profile corresponds to integrating the magnitude-squared of the output of the STFT over all frequencies. These profiles are matched against the predictions to determine if they bear out the assumption that all tones found in the previous analysis interval continue unchanged through the current analysis interval.

Discontinuities are identified in the temporal profile to find possible locations where MNBI tones may have terminated or started. These discontinuities are discrepancies with respect to the predictions which assume that the tones continue unchanged from the previous block to the current one. The time locations of discontinuities are found through a chi-squared test [15].

In the case of the spectral energy profile, discrepancies are found by identifying each spectral region that has significant energy even in the absence of any predicted MNBI tones in that region. Such regions correspond to possible starts of new MNBI tones in the signal. The process of finding such discrepant regions involves first picking peaks in the spectral profile. If any peaks are found in frequency regions that do not overlap with the predicted MNBI tones, then such peaks mark the locations of discrepancies.

Discrepancy Detection in Stage 2:

This mechanism is used to determine whether the TF trajectory of an MNBI tone with FM has been sufficiently refined. The discrepancy detection process compares the TF trajectory obtained using the adjusted analysis filters against the TF trajectory obtained in the previous iteration of the second stage. If the average change between the two trajectories is more than 5% over the entire duration of the MNBI tone, then a discrepancy is declared.

5.3.5 Reprocessing Planning and Reprocessing

The reprocessing planning and reprocessing mechanisms also have two different versions. During Stage 1 the first version is invoked to reprocess TF regions where predictions do not match the tone TF trajectories found in the results of initial STFT processing. The analysis filters are chosen to have sufficient frequency resolution to separate tones that are at least 16 Hz apart. In Stage 2 the second version of the reprocessing planning and reprocessing mechanisms is used to reprocess MNBI tones with FM. In particular, these mechanisms help choose a set of analysis filters that result in accurate TF trajectories for MNBI tones with FM.

5.4 Implementation

We have implemented a C++ software testbed based on our approach to tone separation. This testbed is built on top of the recently introduced *IPUS* C++ Platform (ICP) [17].

In implementing our testbed ICP was used to develop, compile, debug, and test all the components required by the *IPUS*-based framework described in a later section . Using ICP's library of base C++ classes, we derived a version of the blackboard database that is suitable for our spread spectrum application. Functional blocks for the various *IPUS* mechanisms were developed, compiled and tested by taking advantage of the inherent support that ICP provides for C++ software development. Furthermore, we also utilized ICP's libraries to instantiate a RESUN control planner for runtime scheduling of *IPUS* mechanisms. Although a large portion of RESUN's capabilities are not exercised by our application, we used it to provide modularity to the testbed as well as to allow future system upgrades.

Our 4-level blackboard database was derived from ICP's base classes that have a rich set of default behaviors. The different levels of the database are: waveform, spectrum, peaks, and tracks. For each level, the base classes were also used to customize the behaviors of

data objects. For example, the track level data objects permit searching for all tracks that fall within a specified TF region. Inbuilt default behaviors for the base classes also allowed the instantiated data objects to be easily accessed by different $IPUS$ mechanisms during system operation.

We were able to easily implement the various $IPUS$ mechanisms using ICP's scripting language. By using the C++ pre-processor's macro facility, this language facilitates the derivation of software components through natural and direct statements. The language's other advantages include enhanced code readability, reduction in the amount of keyboard entry required for coding a system, and reduction in the amount of C++ knowledge required by the system implementor. Another important mechanism that aids software development in ICP is an interface between the application and external software components. This interface simplified the process of debugging our system and allowed us to display all internal structures and operations of the system in both textual and graphical form at runtime.

To facilitate the proper runtime control of various $IPUS$ mechanisms, we derived a RESUN control planner using ICP's libraries. Although the context-dependent control capabilities of this planner are not fully utilized by our system, we found it to be beneficial in our implementation. The RESUN planner relies on decomposing the system's goals into a goal/plan/subgoal hierarchy. This is illustrated in Figure 5.6, where we show a schematic of the control plans in the reprocessing loop of our application. Decomposition of system goals into such a hierarchy allows modularity during system building. Furthermore, we also anticipate that when our system is upgraded to handle other types of interference (such as wide-band interference), more sophisticated aspects of RESUN will be brought into play.

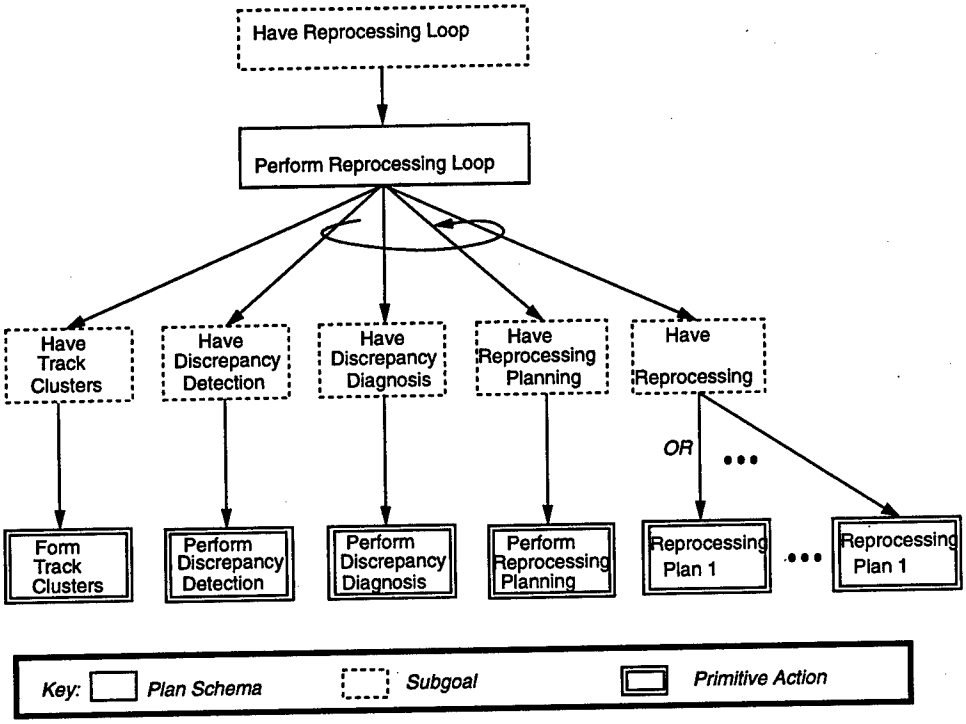


Figure 5.6: Diagram of a portion of the goal/plan/subgoal hierarchy from our testbed.

Chapter 6

Some Prototype Development Issues

An analytical performance evaluation of rule-based systems that rely on knowledge extraction is generally not feasible. In these cases, one needs to employ computer-aided modeling and simulation for system evaluation. In our case, the complexity of the communication system along with knowledge-based interference cancellation makes an analytical performance evaluation extremely difficult. For this reason we employ computer-aided modeling and simulation in order to demonstrate the effectiveness of our approach. To this end we use a signal processing software tool for time-domain modeling and simulation of communication systems called Signal Processing Worksystem ($SPW^{©}$) from Cadence Corp. In this chapter we briefly describe this tool. Another key issue was to establish communication between $IPUS$ and SPW. We also describe our solution to this problem.

6.1 Signal Processing Worksystem $SPW^{©}$

In the signal processing software package SPW systems are completely characterized in the time domain. The spread-spectrum system is modeled in terms of functional blocks such as mixers, amplifiers and detectors. Transmitted signals are generated in the time domain as are the responses of the receiver functional blocks to the received spread-spectrum signal and the interference. This enables us to realistically evaluate system degradation due to

interference and the effectiveness of the proposed rejection scheme.

The signal processing software tool used in this research for the time-domain modeling and simulation of communication systems is the Signal Processing Worksystem (SPW). SPW is an integrated software package for designing signal processing systems. Its vast communication systems and digital signal processing library, graphical design methodology, fast simulator, test and analysis facility, and implementation options make SPW an excellent choice for concept to prototype design. SPW can be used to model and simulate a system, then debug, revise and re-simulate the system until optimal results are obtained.

SPW consists of several modules. The Designer/Block Diagram Editor module is used to graphically represent a system as a set of functional blocks connected by wires. Each block is a symbol that represents an operation and the interconnecting wires symbolize the flow of signals between blocks. The Signal Calculator module is used to create input signals and analyze output signals from the modeled system. It can also perform signal operations such as filtering, convolution, fast Fourier transforms, and cross correlation. The Signal Flow Simulator module is used to convert block diagrams created in the Designer/Block Diagram Editor module into simulation programs. It processes input signals created by the Signal Calculator module and then executes the simulation.

6.2 Custom Coded Blocks

Although SPW comes with a large number of blocks, it is sometimes necessary to create a block which is not provided in the SPW libraries. In this project we needed a few blocks that are necessary to provide a communication capability between SPW and $IPUS$. These blocks are nothing but C codes which carry out certain tasks. Creating a custom coded block can be divided into seven sub-processes.

- Create and Save Symbol: Any block can be created by drawing and labeling the appropriate inputs and outputs. The ports are added to the created blocks in order

to pass data in and out of the blocks.

- Create and Save Parameter Model:

  A parameter screen model is created in order to let the user specify the parameter values. It is created using lines, boxes, and the add parameter command.

- Link Symbol and Parameter Model:

  The created block and its parameter screen file must be linked. By doing so, the block is able to access its assigned parameter values.

- Generate Source Code:

  In the block design window (BDW) there is a command to prepare blocks. When we execute this command, ".c"(C code file) and ".h"(Header file) are generated. The code we write must be put into ".c".

- Insert Code:

  The ".h" file contains all C code structure definitions. It contains all of the procedure call declarations which need to be written.

- Compile Code:

  After having edited the code, it must be compiled. If an error occurs, the appropriate changes are made and the code is recompiled.

- Add New Block to Block-list:

  Before the newly created block is used in any simulation, it must be added to its pool of available blocks.

The following are the functions of some of the important CGS blocks that we have developed in *SPW*:

- I_TRIG_CG : After a block of N samples are stored, this block writes '1' into a certain flag file. This triggers the *IPUS* system to process the stored samples.

- TRIG_4 : After having processed a certain number of samples, The block activates the cancellation engine.
- JAMMER_INFO : This block reads the information about parameters of jamming signals from $IPUS$ system. It send this information to the cancellation engine.
- CW : This block reads information about the time-frequency tracks from $IPUS$ and sends this information to the cancellation engine.

6.3 Integration of SPW with IPUS

Performance evaluation of the knowledge-based interference cancellation scheme presented in this report involves simulating the spread-spectrum system using SPW in conjunction with the $IPUS$ C++ platform. The collaboration of the SPW signal flow simulator and $IPUS$ during the simulation process is an important issue. This can be further explained as follows. The simulation process can be viewed as consisting of three consecutive stages, each one depending on the successful completion of the preceding stage. The first stage consists of the spread-spectrum transmitter and the channel. The transmitter generates the direct-sequence spread-spectrum signal. Disturbances such as AWGN and interference, are introduced in the transmitted signal during the passage through the channel. The interferers are selected from the list of interferers discussed in a previous chapter. A graphical user interface is provided that allows the user to select any combination of interferers from a preselected menu by clicking on the appropriate switch. The spread-spectrum bandwidth is divided into four sub-bands and, in each sub-band, the user is given a choice of selecting an interferer from the menu. This graphical interface could be easily modified to include many more interferers in each sub-band and also to allow for the overlapping in frequency of interferers.

The received signal consists of the transmitted spread-spectrum signal and the interference. The second stage of the simulation consists of processing the received signal by

IPUS. In the third stage interference cancellation is carried out based on the information on the interference extracted by *IPUS*.

The *IPUS* C++ platform has to be integrated with both the *SPW* simulation system that performs the signal and interference generation tasks and the functional stages that perform the interference cancellation and demodulation tasks. This issue is complicated due to the fact that the *IPUS* C++ platform cannot be directly incorporated as an *SPW* custom-coded block since the current version available at the time of development, version 3.5 of the Signal Processing Worksystem, did not support a C++ compiler. An alternative approach is ,therefore, required to pass the received data to *IPUS* for processing and to collect the information extracted by *IPUS* regarding the interference that is disrupting the received spread-spectrum signal. As described earlier in this chapter, *SPW* runs the simulation in the time-domain. The *SPW* scheduler controls the progress of the simulator by passing tokens from one or more blocks that concurrently perform their signal processing tasks to the succeeding block or blocks. Multirate blocks like up-samplers and down samplers are also handled using this token-passing approach. *SPW* facilitates the control of the execution of a particular functional block by means of a 'hold' control signal. The 'hold' signal, when set high, suspends the execution of the block. This feature is used in our simulation in generating data arriving at the spread-spectrum receiver (stage 1), processing of the received data by *IPUS*, and performing the subsequent interference cancellation operations based on the information provided by *IPUS*. We pass data in a block-wise fashion from one stage to the succeeding stage. The length of the block of data is selected such that *IPUS* receives adequate data samples in the block to perform its interference isolation task efficiently without introducing excessive delays due to block-wise processing of data.

The following procedure is adopted for the combined *SPW-IPUS* simulation.

(1) The transmitter generates the DSSS signal and interference is introduced during transmission through the channel. A block of N received signal samples are written to a

data file. *IPUS* is currently in an idle state, and is polling a flag for the signal to start processing the received data. Stage 3 blocks, comprised of the interference cancellation stage and the SS demodulator, are also currently disabled by means of hold-control. As soon as stage 1 has written N data samples to the file, the flag is set to trigger the next stage (*IPUS*) to start processing data. Once *IPUS* is triggered, stage 1 goes into the idle-state and waits until the succeeding stages have completed processing the current block of N data samples. Enabling and disabling the blocks in stage 1 is achieved by setting the hold signal to either a low or high value, respectively.

(2) The *IPUS* data processing cycle is triggered by the flag. *IPUS* resets the *IPUS*-trigger flag and starts processing the received data. The information transfer from *IPUS* to stage 3 is also done through the Unix system interface. *IPUS* generates time-frequency tracks for each identified jammer and writes the information in file(s). Stage 3 is still in the idle state and is polling the 'stage-3 trigger' flag to determine when to start reading the processed information from *IPUS*. Once *IPUS* has completed processing the block of N received data samples, the stage-3 trigger flag is set and *IPUS* goes back to the idle state and keeps polling the IPUS-trigger flag to determine when to start reading the next block of samples.

(3) Stage 3 is triggered by *IPUS* by setting the stage-3 trigger flag. Once triggered, stage 3 resets the stage-3 trigger flag and proceeds to read the jammer information written in data files by *IPUS*. The stage 3 processing cycle continues until the current N data samples have been processed. The stage 3 signals to stage 1 by setting the flag that enables the hold-signal for stage 1 blocks. Upon detecting the flag, Stage 1 resets the flag and starts generating the next block of N data samples at the receiver.

A control block is developed in stage 1 for the purpose of generating the control signals that trigger the next stage, and the hold signal for stage 1. A custom-coded block is designed for the purpose of polling and resetting the flag set by stage 3. Figures 6.1 and 6.2 show the circuits that generates the trigger signals for *IPUS*.

The process of writing to and reading from data files through the Unix interface during the simulation increases the overhead for the simulation program. However, this approach is only utilized for demonstrating the effectiveness of our knowledge-based interference cancellation scheme. The execution time for this combined SPW-$IPUS$ simulation could be significantly reduced in the near future with the release of SPW Version 4.0 which supports a C++ compiler. This will enable $IPUS$ to be directly incorporated into the SPW system by means of encapsulating $IPUS$ with an SPW shell and representing $IPUS$ as a C++ custom coded block. The same block-wise processing of data could be performed without having to transfer data and control signals through the Unix system interface.

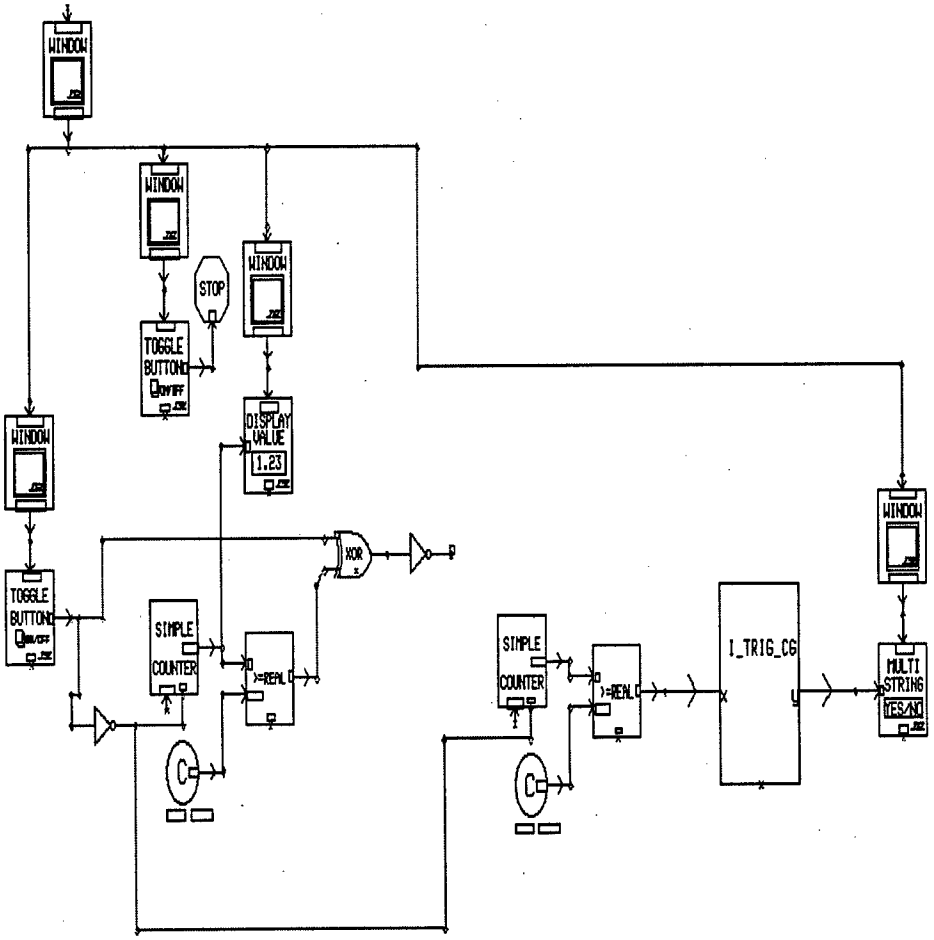


Figure 6.1: Integration Circuits Located at the Transmitter

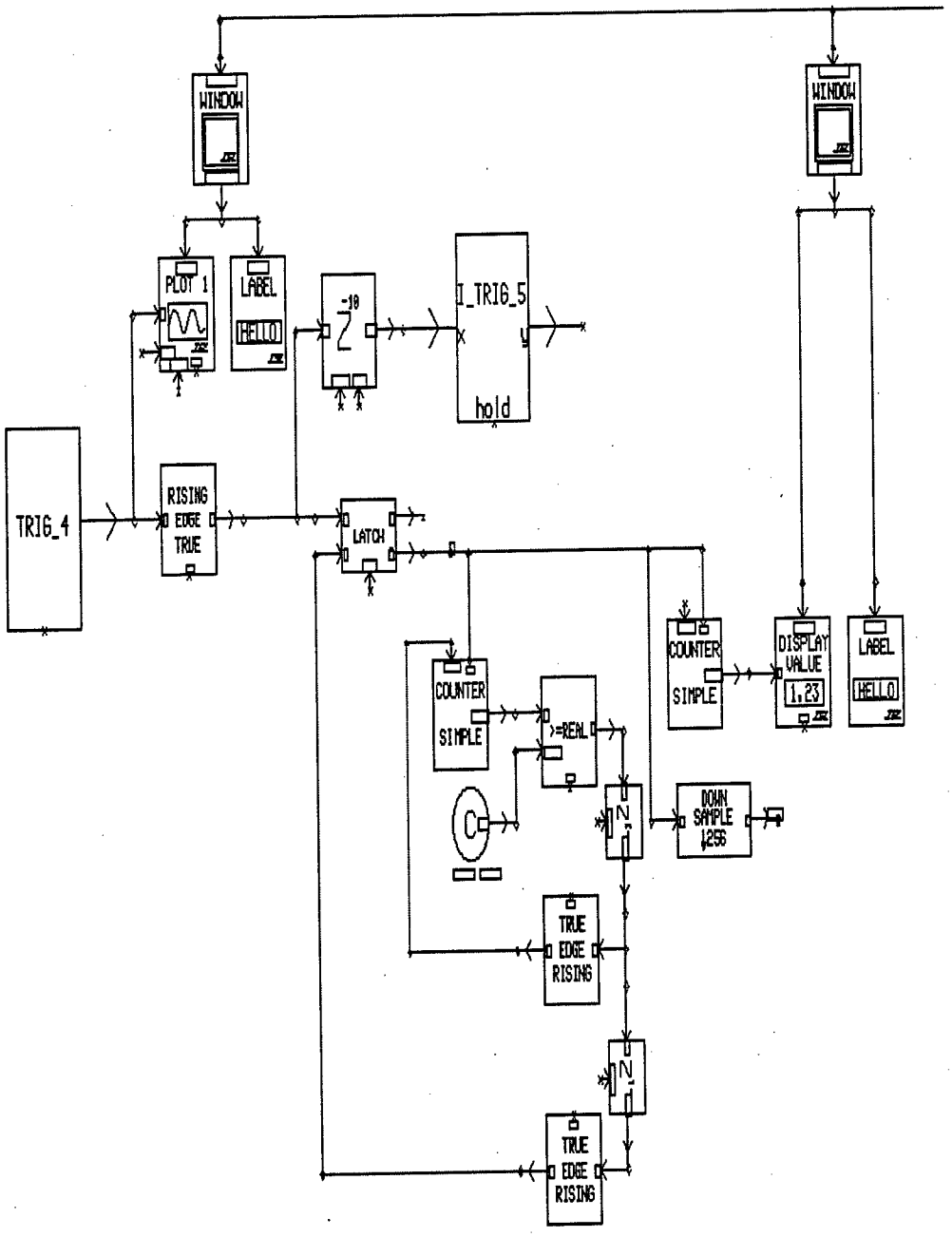


Figure 6.2: Integration Circuits Located at the Receiver

Chapter 7

System Models and Experimentation

7.1 System Models in SPW

The main focus of this chapter is to present models for communication system blocks that have been created in *SPW* to demonstrate the effectiveness of our knowledge-based interference cancellation approach. We also include some discussion on the development of rules for the selection of interference cancellers from the library of filters.

7.1.1 Transmitter

The model for the spread-spectrum transmitter is shown in Figure 7.1. The random data generator generates 0 and 1's with equal probability and provides the information bits for the BPSK modulator. The spreading signal is generated by the PN sequence generator whose values of 0 and 1 are transformed to -1 and +1, respectively, by the block designated as binary to numeric. The bit rate of the PN codes is 32 times that of the information bit rate. This provides the spread-spectrum system with a processing gain of 32 or 15 dB. The spread information bit stream is modulated by a carrier prior to transmission over the channel. The parameter values for the transmitter are listed in Figure 7.2. Typical spectra of the input message and transmitter output are shown in Figure 7.3.

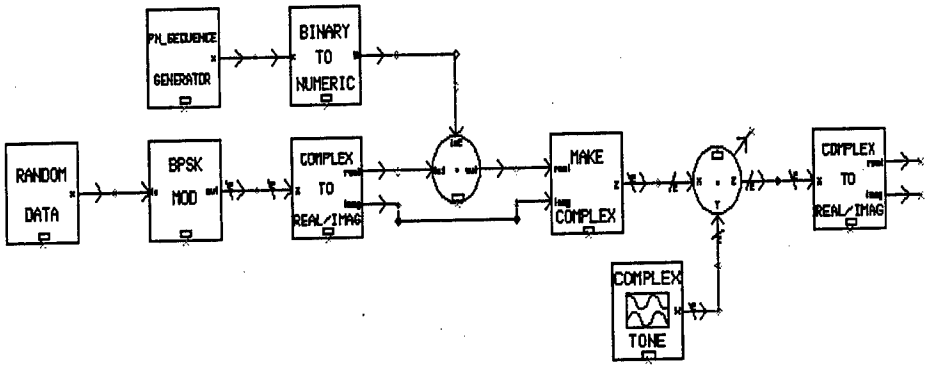


Figure 7.1: Detailed Model of the Transmitter in SPW

Transmitter Gain	1.0
Data Rate	100 bits/sec.
PN Code Rate	3200 bits/sec.
Sampling Rate	16384 Hz.
Carrier Frequency	4800 Hz.

Figure 7.2: Transmitter Parameters

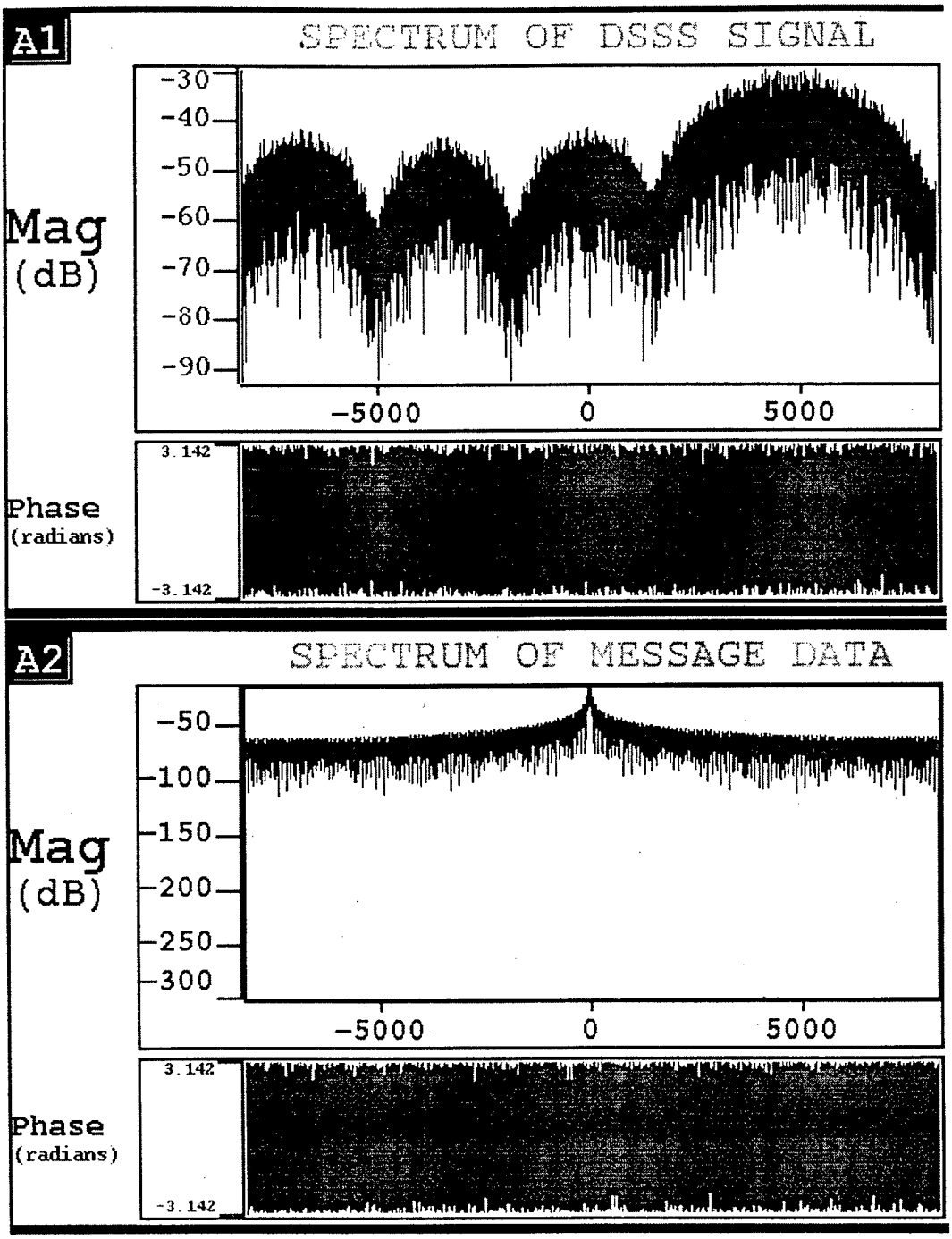


Figure 7.3: Spectra of the Transmitted Signal and the Input Message

7.1.2 Receiver

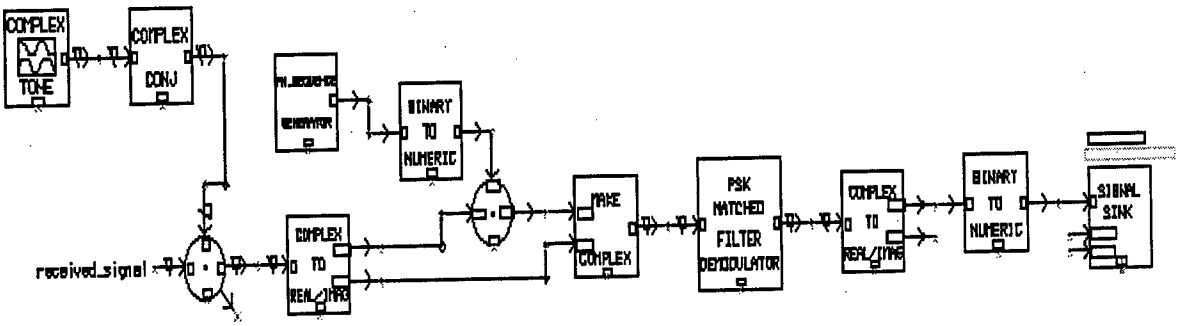


Figure 7.4: Detailed Model of the Receiver in SPW

Receiver Gain	**1.0**
Baud Rate for Demodulation	**100 bits/sec.**
PN Code Rate	**3200 bits/sec.**
Sampling Rate	**16384 Hz.**

Figure 7.5: Receiver Parameters

Figure 7.4 shows the design of the receiver portion of the system. The received signal at the receiver input is down-converted to baseband by multiplying with the conjugate of the carrier used at the transmitter. The resulting signal is despread by multiplying with the PN code which is a replica of the PN code used at the transmitter. The despread signal is passed into a matched filter demodulator block which detects the information bits. This block integrates over each symbol interval and quantizes the output to the nearest PSK constellation point in order to reproduce the digital information. The binary to numeric

block converts -1 and +1 at the demodulator output to 0 and 1, respectively. The parameter values for the receiver are listed in Figure 7.5.

7.1.3 Channel and Interference Signals

During transmission over the channel the transmitted spread-spectrum signal is contaminated by additive white Gaussian noise (AWGN) and jamming signals. Figure 7.6 shows the block diagram of the channel. In the channel model interference signals and AWGN are added to the transmitted spread-spectrum signal. At any given time one or more types of jamming signals can be added to the channel. The following paragraphs describe how the jamming signals were implemented using *SPW*.

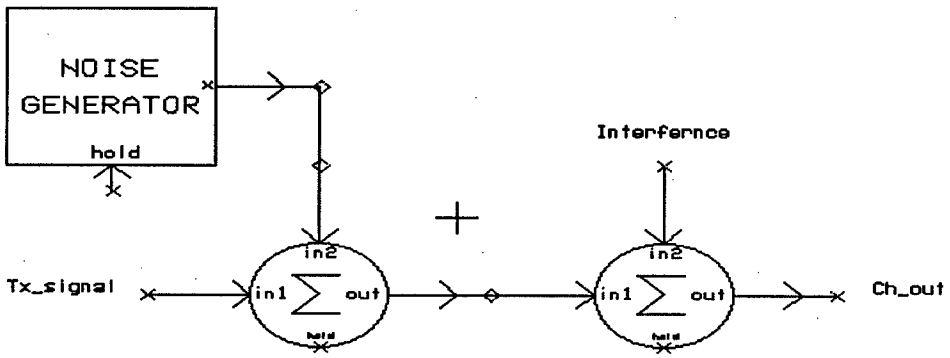


Figure 7.6: Channel Model

7.1.4 Linear FM (LFM) signal

Figure 7.7 shows the model of a subsystem that generates a LFM jammer signal which can sweep over the desired bandwidth. The bandwidth and the center frequency of the band are specified from the user-interface. When a triangular signal, which is obtained from the output of the first signal generator, is applied to the frequency input of the second signal generator, the output of the second signal generator is a linear FM (swept) jamming signal. The sweep rate is controlled by signal parameters. The power of the the LFM jammer signal is fixed by multiplying with a suitable constant. Figure 7.8 illustrates the spectrum of a

direct-sequence spread spectrum signal contaminated by a swept jammer sweeping over a limited portion of the transmitted signal spectrum. In Figure 7.7 the window, scroll bar and display value blocks provide for the user-interface.

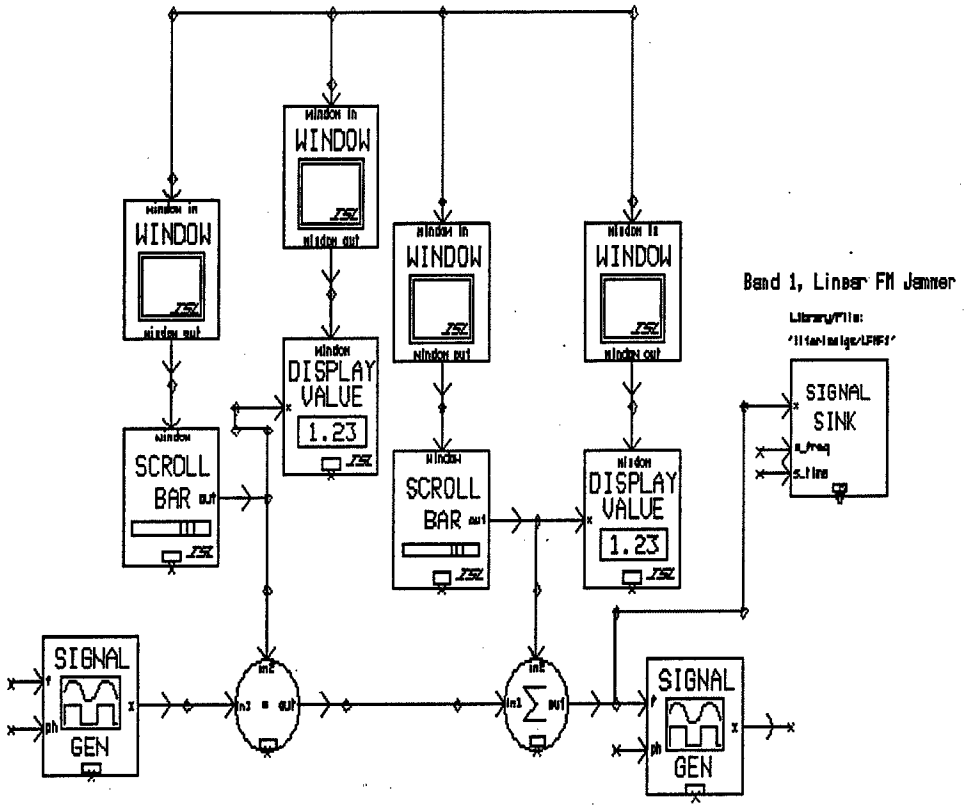


Figure 7.7: Detailed LFM Jammer Model in SPW

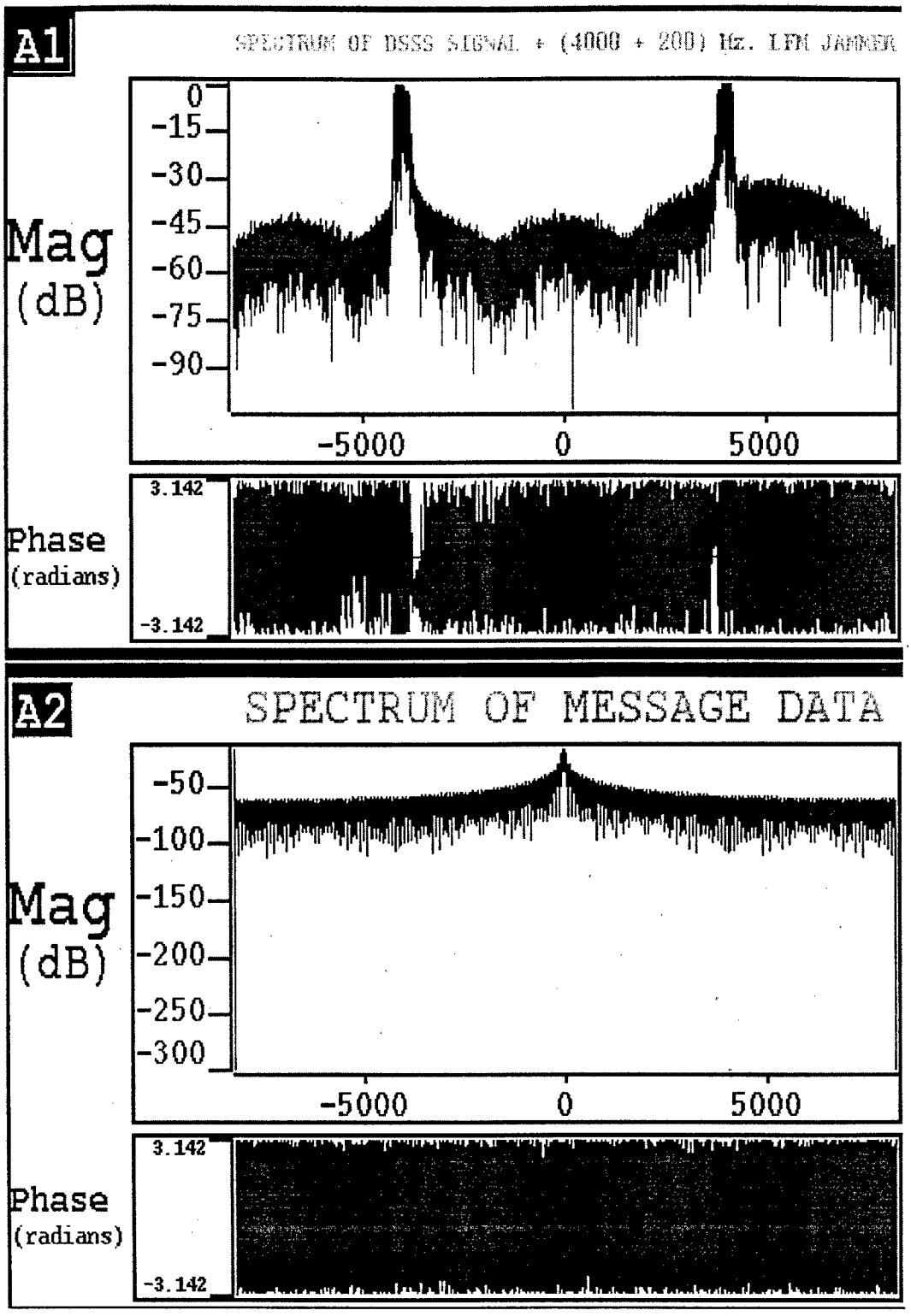


Figure 7.8: Spectra of the Transmitted Signal Plus LFM Jammer

7.1.5 CW jammer

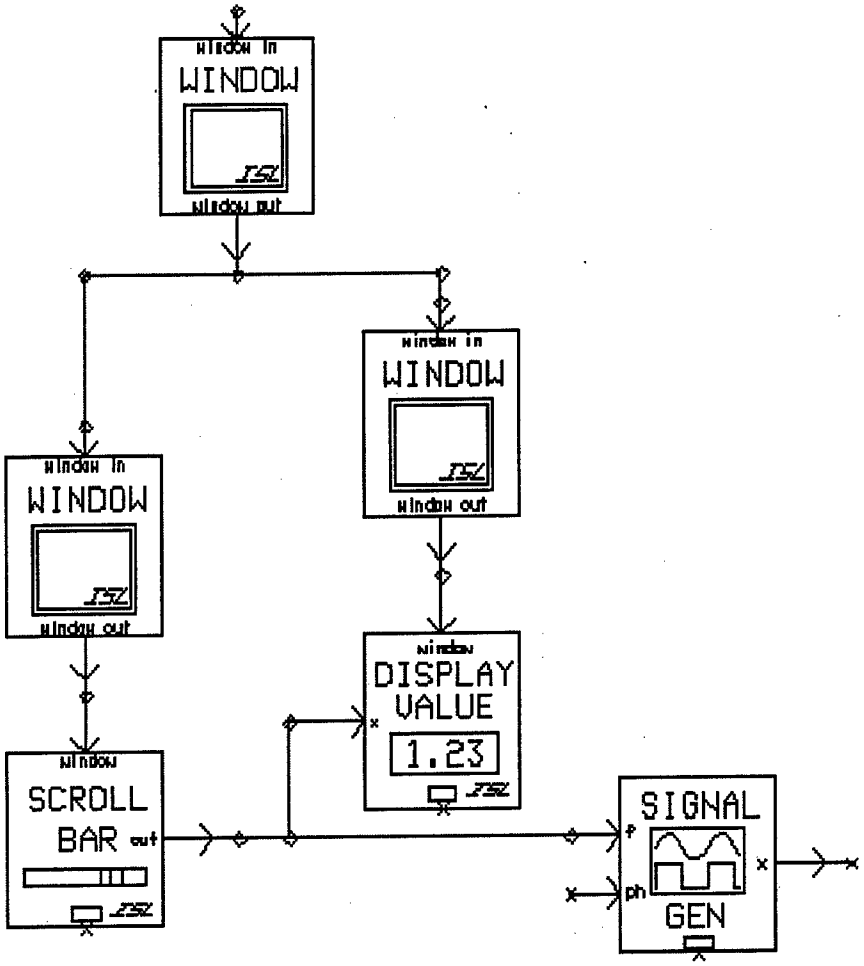


Figure 7.9: Detailed SPW Model of the CW Jammer

The simple design module shown in Figure 7.9 generates a CW jamming signal. By using the user interface, the CW jamming signal can be positioned at a frequency within the spectrum of the transmitted signal. The power of the CW jammer is adjusted by multiplying the output with a suitable constant. Figure 7.10 shows the spectrum of a DSSS signal contaminated by a CW jammer. Once again the window, display value and scroll bar blocks of Figure 7.9 provide for the user-interface.

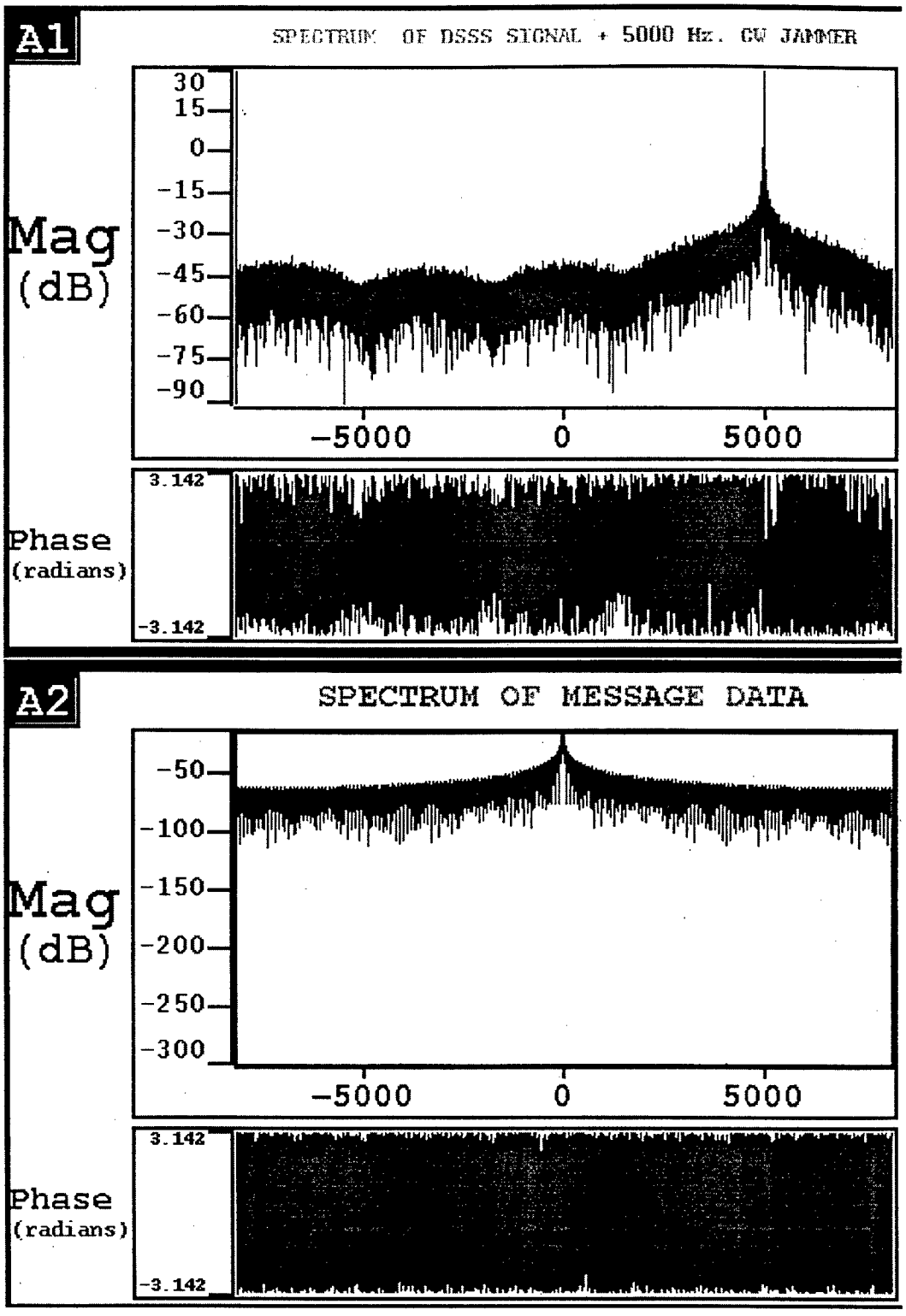


Figure 7.10: Spectrum of DSSS Signal Plus a CW Jammer

7.1.6 Random ON/OFF Jammer

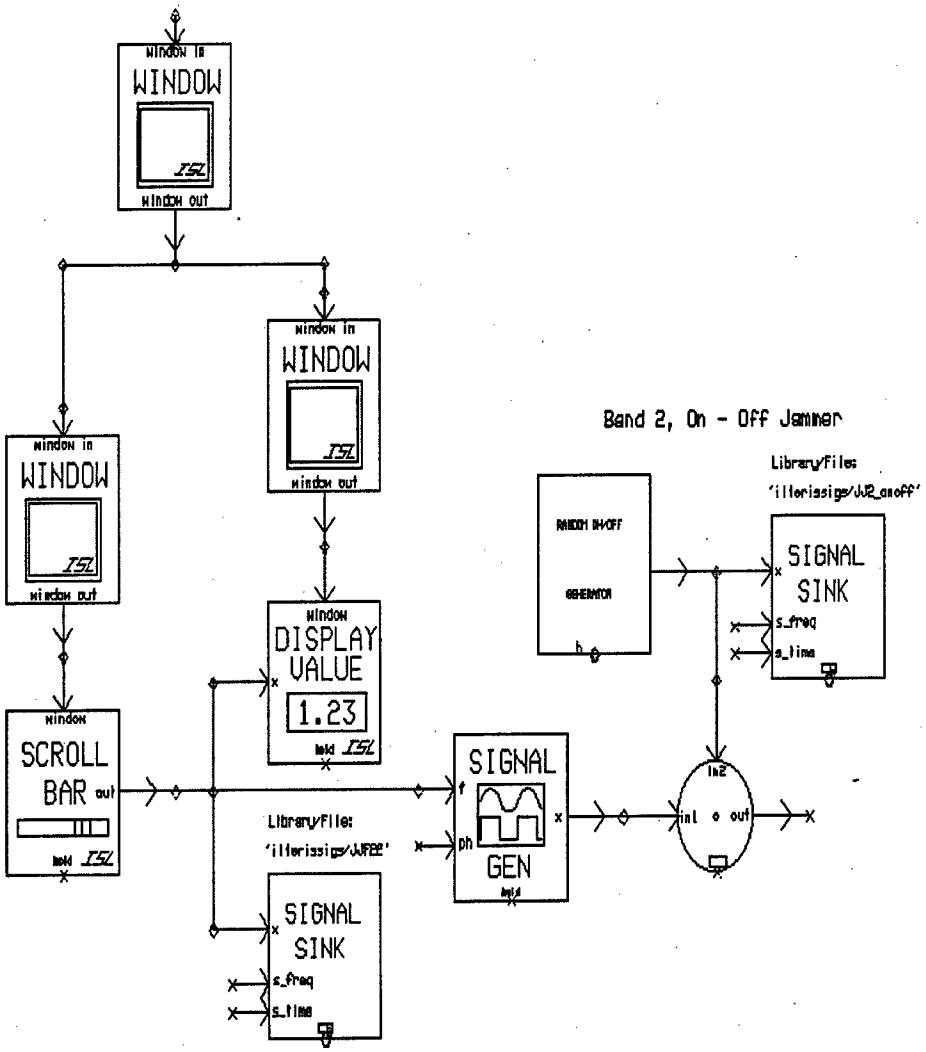


Figure 7.11: Detailed SPW Model of ON/OFF Jammer

The design module shown in Figure 7.11 generates a CW signal which is randomly turned ON and OFF. It is generated by multiplying the CW signal generator with either zero or unity, as determined by the random ON/OFF generator. The frequency of ON/OFF jamming signal can be positioned within the spectrum of the transmitted signal. As before, the window,display value and scroll bar blocks are used for the user-interface.

7.2 PLL Approach for Interference Cancellation

Although *SPW* includes a PLL model in one of its libraries , we had to design our own PLL, because the standard *SPW* PLL block provides only one input whereas two inputs are needed for our system. The first input is for the conventional input signal while the second is for the estimated CW jammer frequency at which the PLL is to be locked. Our PLL design has three basic components:

- A phase detector
- A low-pass filter (Lead-Lag Filter)
- A voltage-controlled oscillator (VCO).

The VCO block is an oscillator that produces a periodic waveform whose the frequency may be varied about some free-running frequency according to the value of the applied voltage. The VCO frequency is the free-running frequency when the applied voltage is zero. The second PLL input is used to set the free-running frequency. The phase detector produces an output signal that is a function of the phase difference between the incoming signal and the VCO block output signal. The resulting signal is filtered using the lead-lag filter block (low-pass filter) in order to get the control signal that is needed to change the frequency of the VCO output. The frequency of the VCO converges to that of the CW jammer. In effect, the VCO output signal becomes the CW jamming signal to within a constant amplitude. If the interference-to-signal power ratio is large enough and the interference is sufficiently narrow-band, the PLL achieves phase-lock in the presence of the desired signal and thermal noise. When the amplitude of the low-pass filter block output is dominated by interference , the interference amplitude, as explained in chapter 3, can be estimated from the low-pass filter output. By multiplying this estimate with the output of the PLL circuit, the desired estimate of the interference signal is obtained. The estimated interference signal is subtracted from the received signal to recover the transmitted signal. Figure 7.12 shows the PLL CW interference canceller model.

In Figures 7.13 and 7.14 we show the performance of the PLL-based canceller. As seen from Figure 7.14, an error in the estimate of the CW jamming signal frequency results in many errors in the data output. In general, the PLL-based cancellation technique does not work well with other types of jammers such as LFM and wide-band jamming signals.

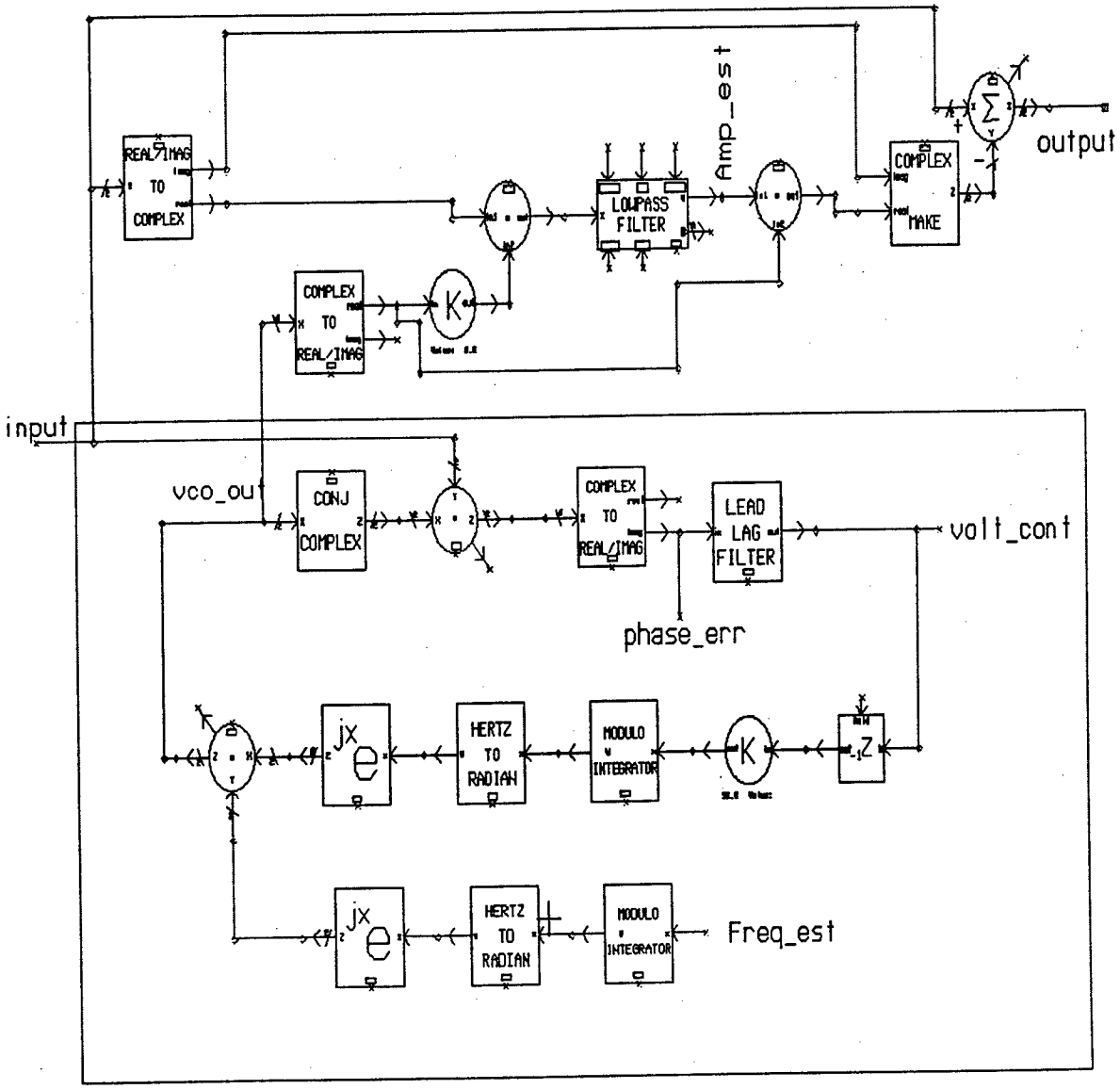


Figure 7.12: Detailed SPW Model of the PLL Interference Canceller

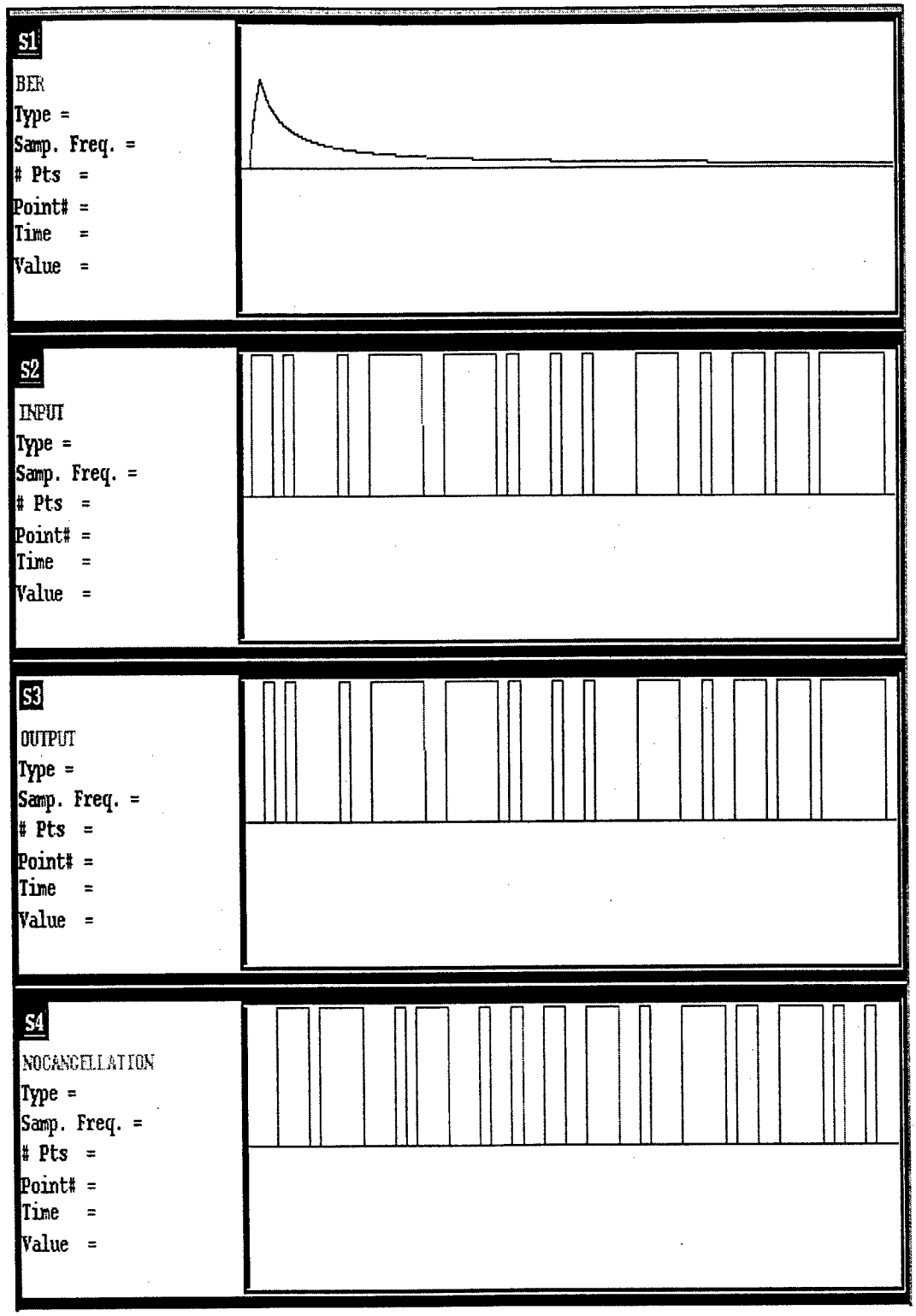


Figure 7.13: Performance of the PLL-based CW Canceller When a Good Estimate of the CW Jamming Frequency is Used at the Second PLL Input.

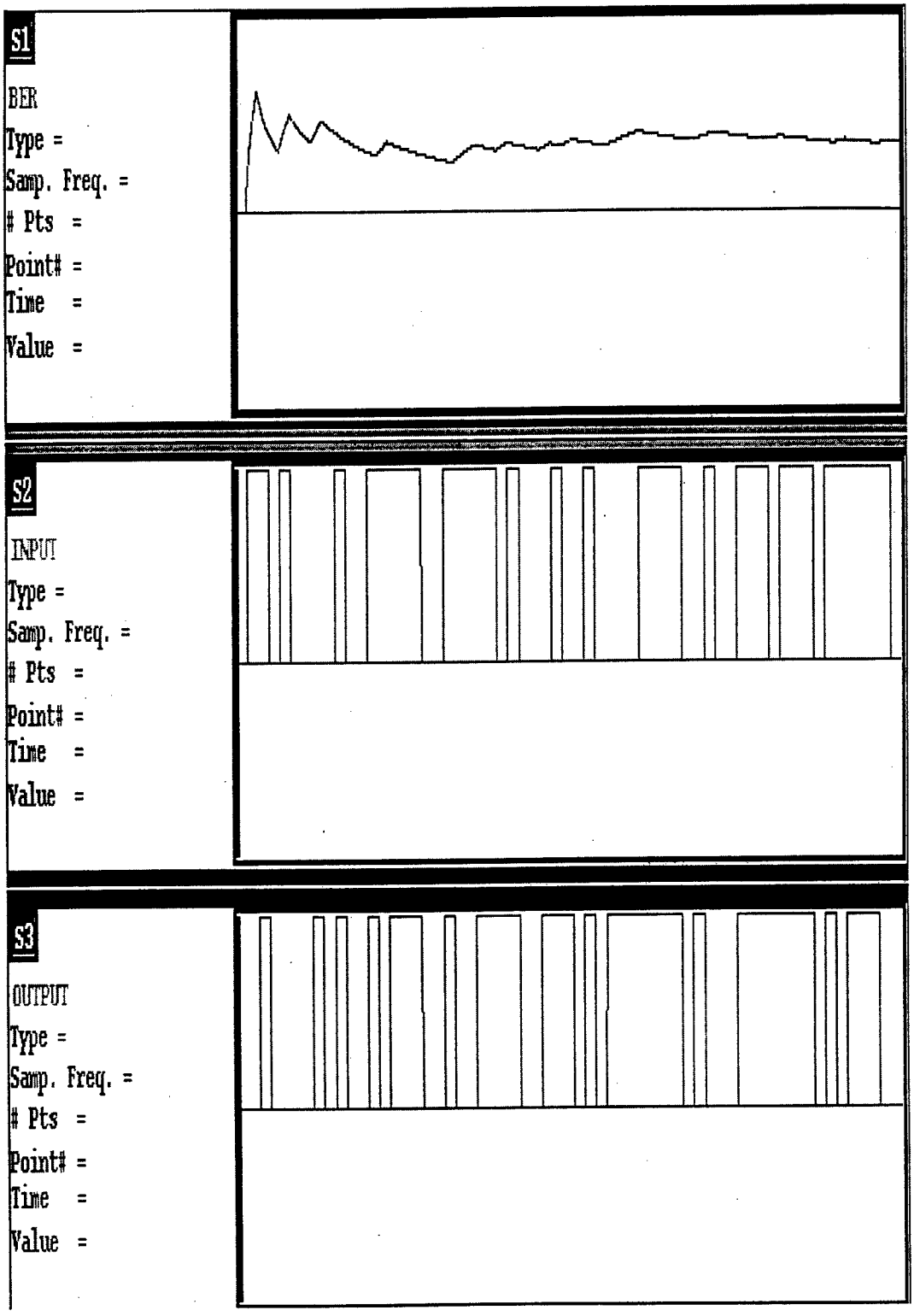


Figure 7.14: Performance of the PLL-based CW Canceller When a Poor Estimate of the CW Jamming Frequency is Used at the Second PLL Input.

7.3 Transform Domain Excisor

The *SPW* design for the transform domain excisor is shown in Figure 7.15. It is used to reduce the effect of jamming signals on the receiver. First, the received signal is decomposed into its real and its imaginary parts. These components are then converted to 2048 point vectors using the 'Scalar to Vector' blocks. The vectors are passed onto the Forward FFT block. This block windows the data vector and performs an FFT using the conventional formula

$$X(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)e^{(\frac{-2\pi jkn}{N})} \tag{7.1}$$

where X(k) is the FFT of the signal vector x(n), k=0,1,2,...,(N-1) and N=2048 is the length of the vector. In the FFT processing, a window may be selected from Hamming, Bartlett, Blackman or Hanning windows. If the magnitude of the FFT component exceeds a prespecified threshold level, a zero is inserted in the corresponding frequency bins. As a result, jamming signals with very high power are removed. After having inserted zeros in frequency bins with high power, the resulting vector is applied to an FFT block that performs the inverse FFT according to

$$x(n) = \frac{1}{N}\sum_{n=0}^{N-1} X(k)e^{(\frac{2\pi jkn}{N})}. \tag{7.2}$$

The resulting output is a complex signal vector with real and imaginary components. This is converted to a complex scalar time signal using the ' Vector to Scalar' block. The resulting complex signal is then passed through the demodulator. Using this transform domain excision technique, jammer tones are excised while the remainder of the spectrum remains unchanged. The excision operation is illustrated in Figure 7.16 . The CW jammer was sufficiently large to cause significant errors in the demodulated output. As shown in Figure 7.17, perfect demodulation was achieved by employing the Transform domain excisor.

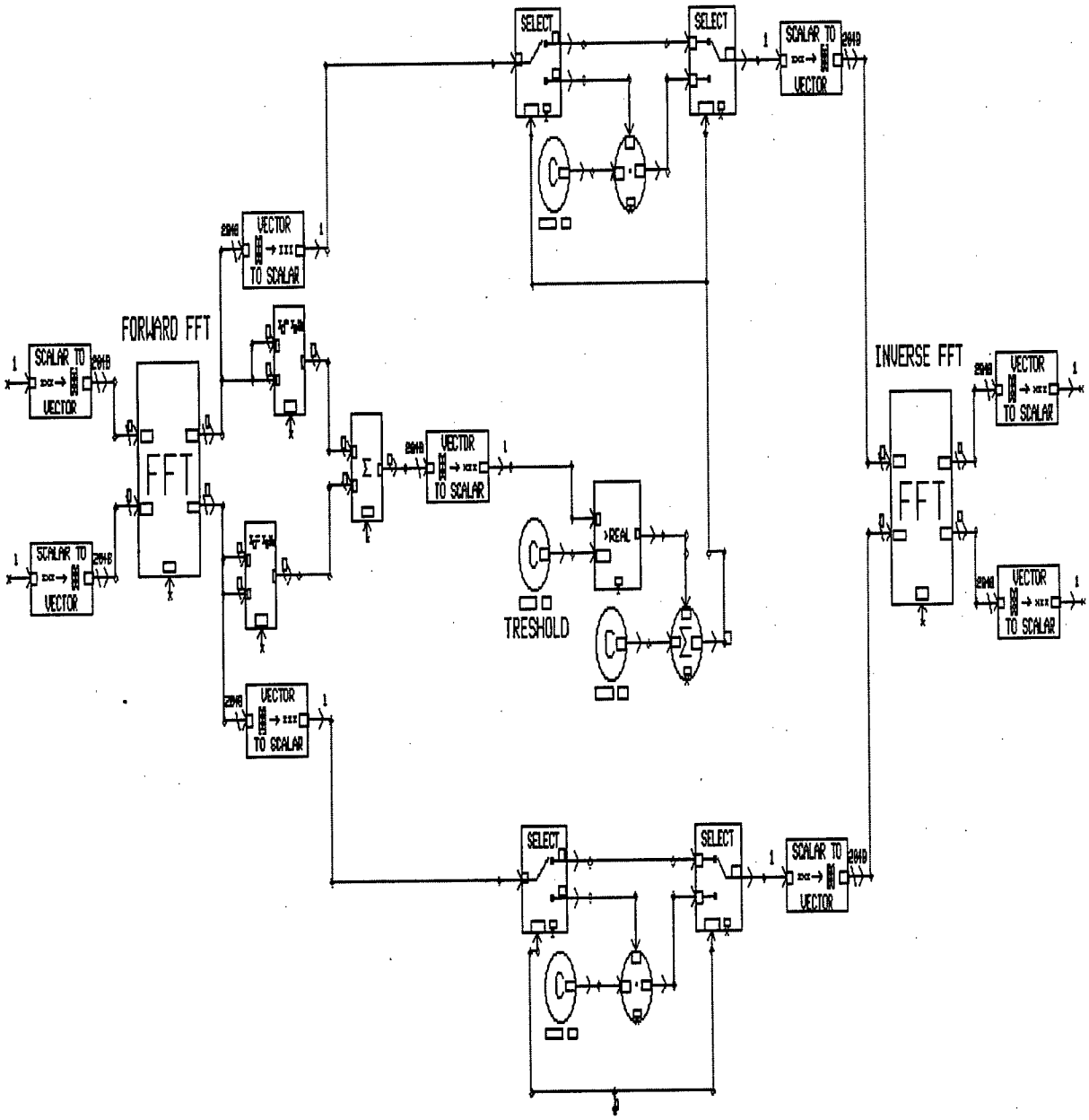


Figure 7.15: Detailed Excisor Model in SPW

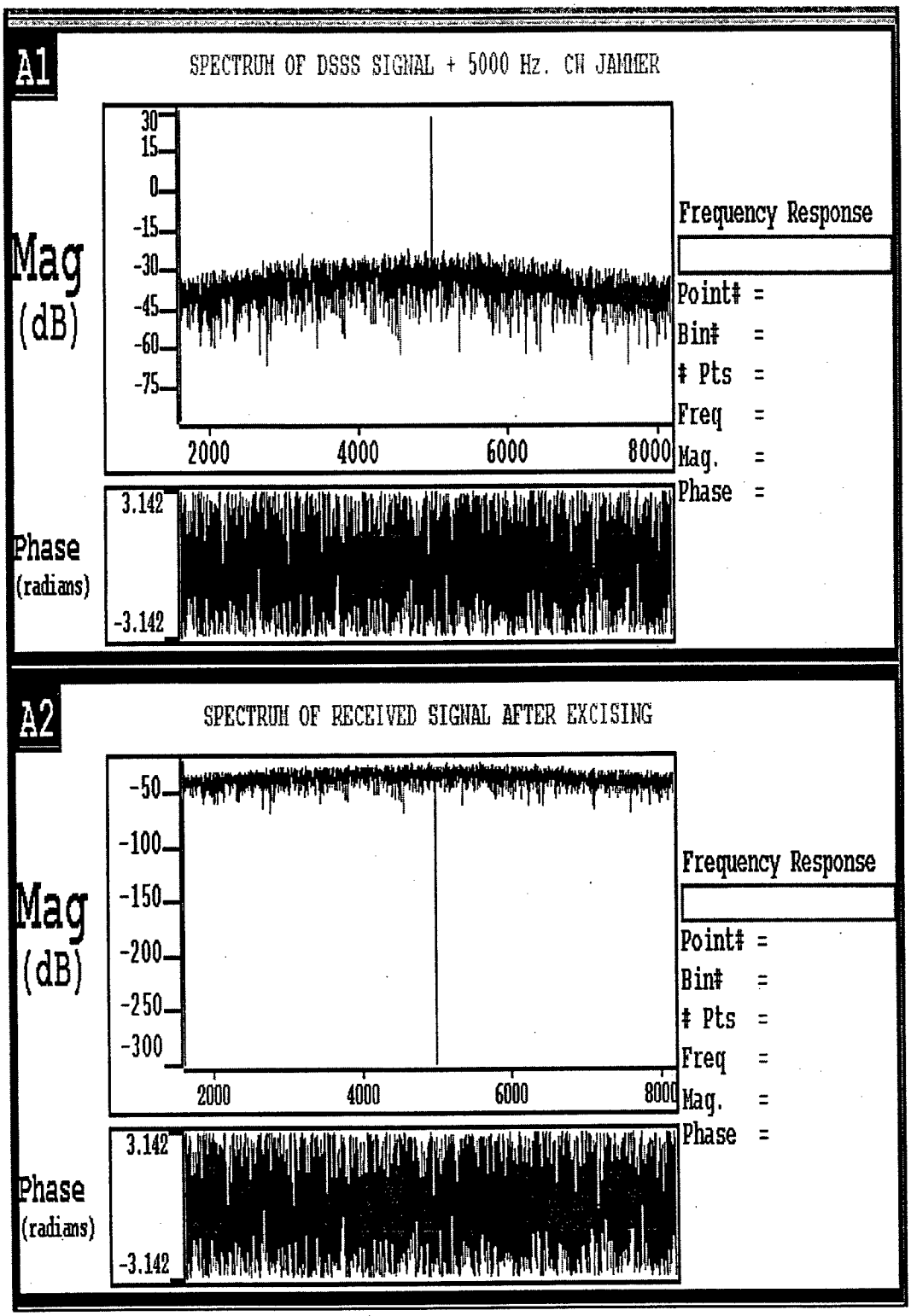


Figure 7.16: Insertion of a Zero at the Jamming Frequency Using an Excisor

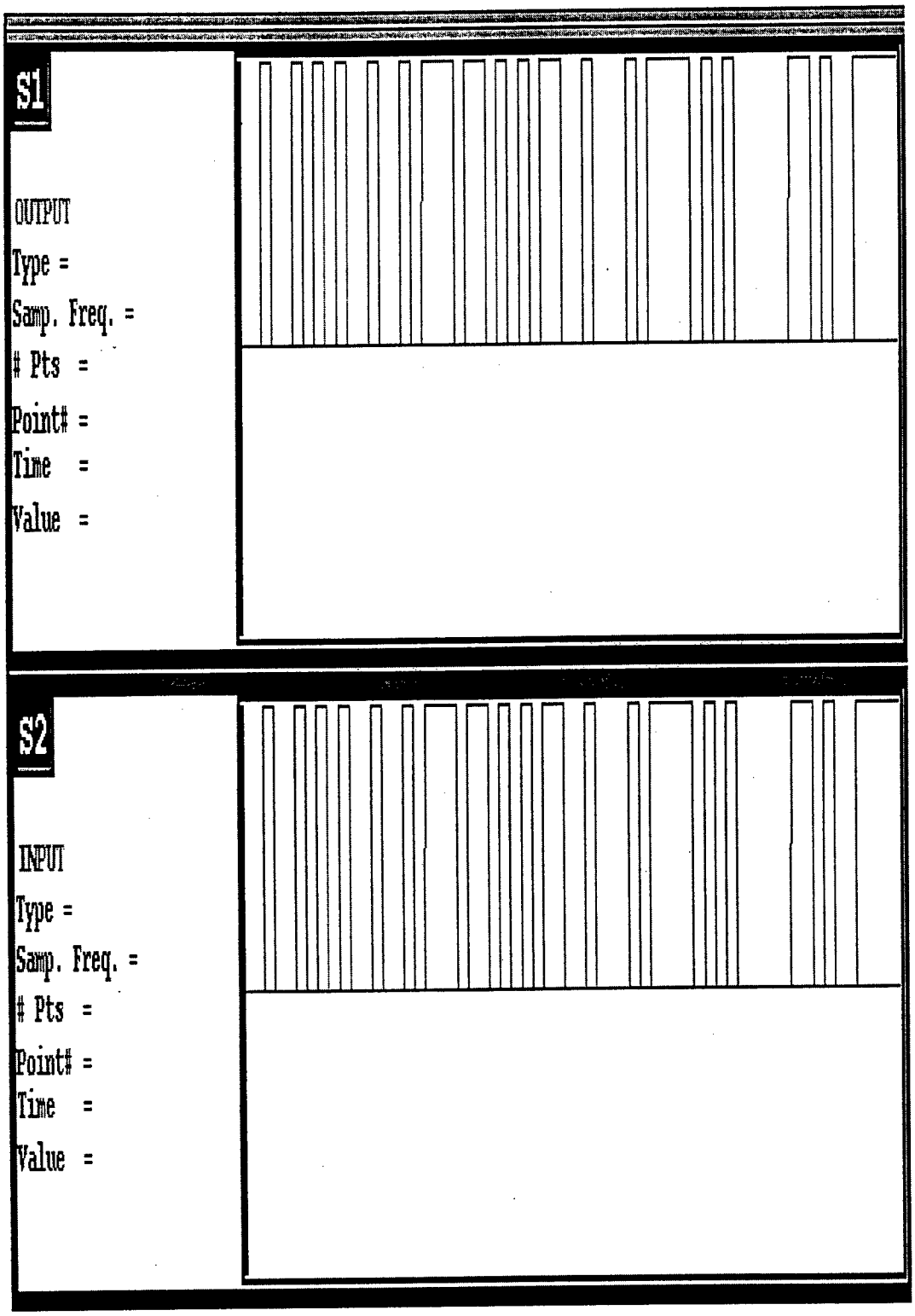


Figure 7.17: System Input and Output when an Excisor is Employed

7.4 Knowledge-Based Interference Cancellation

As indicated earlier, our knowledge-based interference cancellation approach relies heavily on the expert system *IPUS*. This system processes the received signal to extract knowledge such as the presence/absence of interferers, number and types of interferers, their parameters, etc. Based on this information, a set of rules are used to select the most suitable interference cancellation schemes from the library of cancellation algorithms.

In our proof-of-concept demonstration we have limited attention to linear FM (or chirp) jammers and/or single or multiple tone jammers that may or may not operate in an ON/OFF mode. Moreover, we assumed ideal operating conditions in the sense that no other uncertainties exist, e.g. lack of synchronization. The types of jammers considered are completely characterized by their instantaneous frequencies and the times at which they turn ON and OFF. *IPUS* is able to provide this information quite accurately in terms of time-instantaneous frequency tracks. With this limited set of jammers and the information provided by IPUS, it was discovered that simple notch filters are capable of reducing the amplitudes of the jamming signals to acceptable levels. However, the type of notch filter depends upon the type of interference. This considerably simplified the rules needed for selecting the appropriate cancellation scheme. In a more different diverse environment it is expected that other cancellation schemes will be needed and additional rules will need to be developed.

The approach used in the current version of the knowledge-based interference cancellation system prototype is to have the notch of a band-stop filter track the estimated instantaneous frequency of the jammer. The residual interference from the moving notch filter was found to be small enough so that it was readily handled by the processing gain of the spread-spectrum system. The filter coefficients of the notch filter are set to center the notch on the estimated instantaneous jammer frequency. Information provided by *IPUS* in terms of time-instantaneous frequency tracks was found to be adequate for designing the desired moving notch filter.

Although moving notch filters exhibited very good cancellation performance, they did introduce transients that degraded performance. For this reason, it was decided to use the notch filter only when needed. Based on the ON/OFF information provided by the time-frequency tracks received from $IPUS$, notch filter was inserted only when a jamming signal was present in the received signal. Otherwise, the filter was bypassed. This is implemented by means of switches.

The cancellation scheme uses two types of band-stop filters. They are Chebychev Type 2 and the elliptic type band-stop filters. One of the major reasons this filter was chosen for interference cancellation is that it makes the transition between the passband and the stopband very quickly. This feature makes this type of filter an excellent choice, when we receive the transmitted spread-spectrum signal in the presence of slow swept (LFM) jammer.

An elliptic band-stop filter was also used. This filter has an equiripple passband attenuation and the stopband attenuation oscillates between total stoppage and the attenuation at the stopband edge. The smallest attenuation in the stopband is not specified directly but is determined by the other filter parameters. Both the passband and stopband frequencies are specified. The attenuation in the passband oscillates between 0 dB and the value specified at the passband edge. The stopband attenuation is always greater than or equal to its value at the stopband edge.

Based upon the information obtained from $IPUS$, the Chebychev Type 2 filter was chosen to handle CW or a swept (LFM) jammers. However, the elliptic band-stop filter was selected to combat ON/OFF CW jammers. A diagram of the interference cancellation engine is shown in Figure 7.18. The effectiveness of this approach is demonstrated in the next chapter.

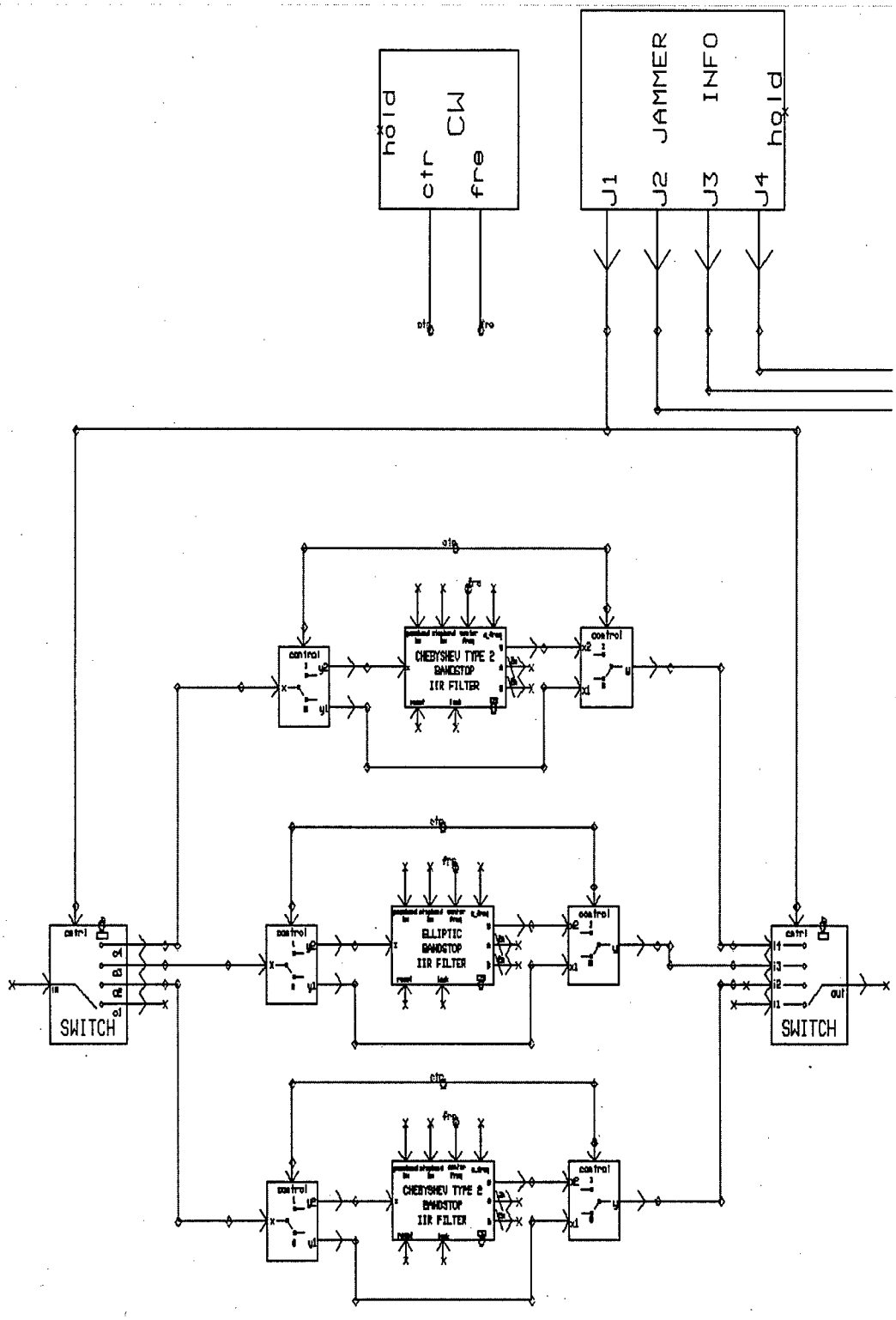


Figure 7.18: Interference Cancellation Engine Implementation

Chapter 8

Simulation Results

This chapter presents the results of simulations for three different scenarios to demonstrate the performance improvement achieved by our knowledge-based interference cancellation scheme over that of the adaptive transversal filter. Performance is characterized in terms of the bit error rate (BER). The BER curves were obtained assuming perfect synchronization. Different combinations of jamming signals are used in the three scenarios where the power of each jammer is of a magnitude sufficiently large to over ride the processing gain of the spread-spectrum system so as to result in significant errors at the demodulator output.

8.1 Scenario 1:

This scenario has four simultaneous CW jamming signals contaminating the transmitted spread-spectrum signal. The frequency of each CW jamming signal falls within the spectrum of the transmitted signal. Figure 8.1 illustrates the time-instantaneous frequency track of these jamming signals. The power of each CW jamming signal is greater than 15 dB above the power of the transmitted signal. Upon analyzing the interference corrupted received waveform, the *IPUS*-based interference isolation stage indicated the presence of four CW jamming signals and provided time-frequency tracks for each of the interfering signals. In Figure 8.2 we show the estimated time-instantaneous frequency tracks. The

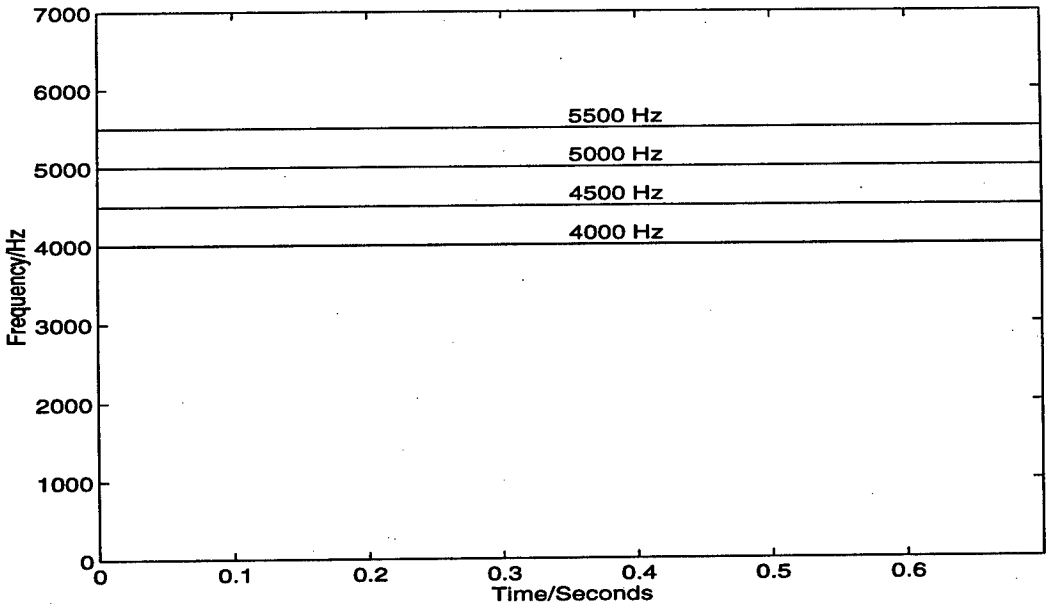


Figure 8.1: Actual Time-Frequency Distribution of the Four CW Jamming Signals in Scenario-1

frequency estimates are then used to adaptively center the notch frequency of the band-stop filters. The frequency location and time profile of each interferer are updated every 16 samples in order to avoid having to adjust the band-stop filter parameters too frequently. The error introduced by updating the frequency estimates every 16 samples is compensated for by adjusting the width of the stopband of the band-stop filters. Bit error rates were obtained for both our knowledge-based cancellation scheme and an LMS-based adaptive transversal filter of order 20. The BER curves are shown in Figure 8.3 where it is seen that the knowledge-based cancellation scheme outperforms the adaptive transversal filter.

8.2 Scenario 2:

In this scenario the received signal consists of the transmitted spread-spectrum signal plus two CW jammers and two ON/OFF jammers. The time-frequency distribution of the jamming signals for a short time segment is shown in Figure 8.4. In this time segment the 4500 Hz jammer turns OFF when the 5500 Hz jammer turns ON. However, the ON/OFF

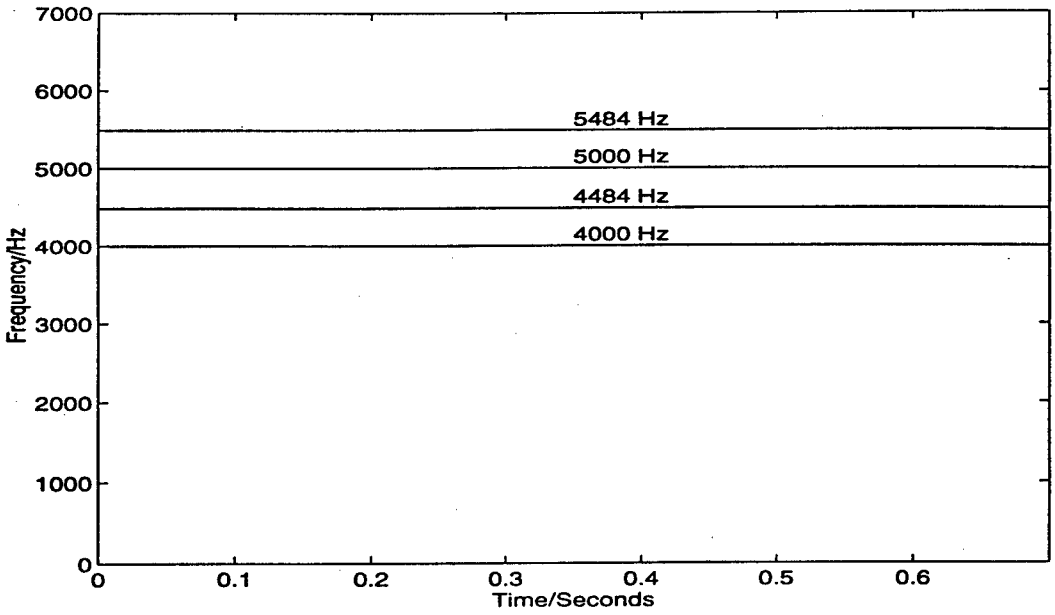


Figure 8.2: Estimated Time-Frequency Distribution of Four CW Jamming Signals Generated by IPUS for Scenario-1

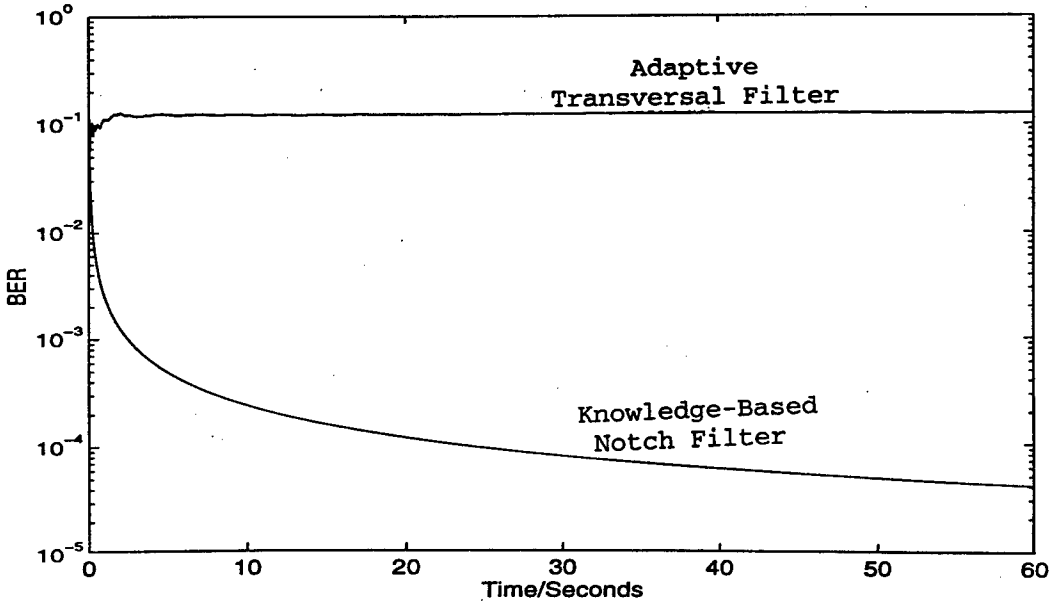


Figure 8.3: Performance Comparison for Scenario-1

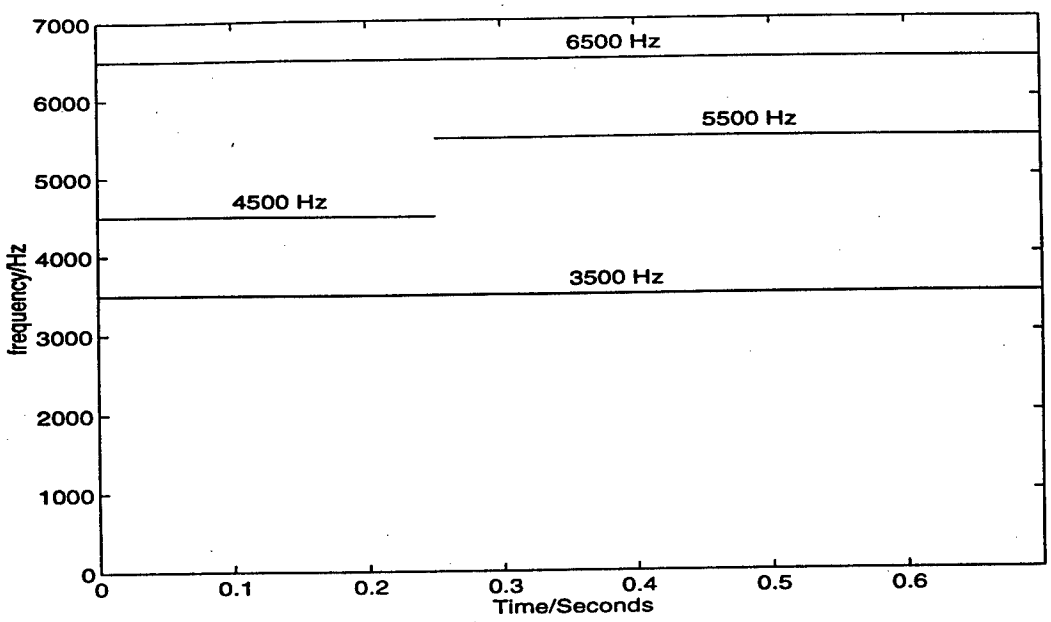


Figure 8.4: Actual Time-Frequency Track of Two ON/OFF and Two CW Jamming Signals in Scenario-2

times are randomly chosen and in other time segments both jammers may be either ON or OFF at the same time. *IPUS* provides the time-frequency locations of each interferer embedded in the received spread-spectrum signal. The time-frequency estimation information shown in Figure 8.5, provides a good estimate of the true time-frequency profile. As with scenario 1, the BER curves are shown in Figure 8.6 for both the adaptive transversal filter of order 20 and the knowledge-based notch filter. As before, the knowledge- based notch filter outperforms the adaptive transversal filter.

8.3 Scenario 3:

This scenario consists of a linear chirp jammer plus three CW jammers in addition to the transmitted spread-spectrum signal. The actual time-frequency track of the LFM jammer is shown in Figure 8.7 along with the piece-wise constant estimate provided by *IPUS*. As shown in Figure 8.8, the knowledge-based notch filter, once again, outperforms the adaptive transversal filter of order 20.

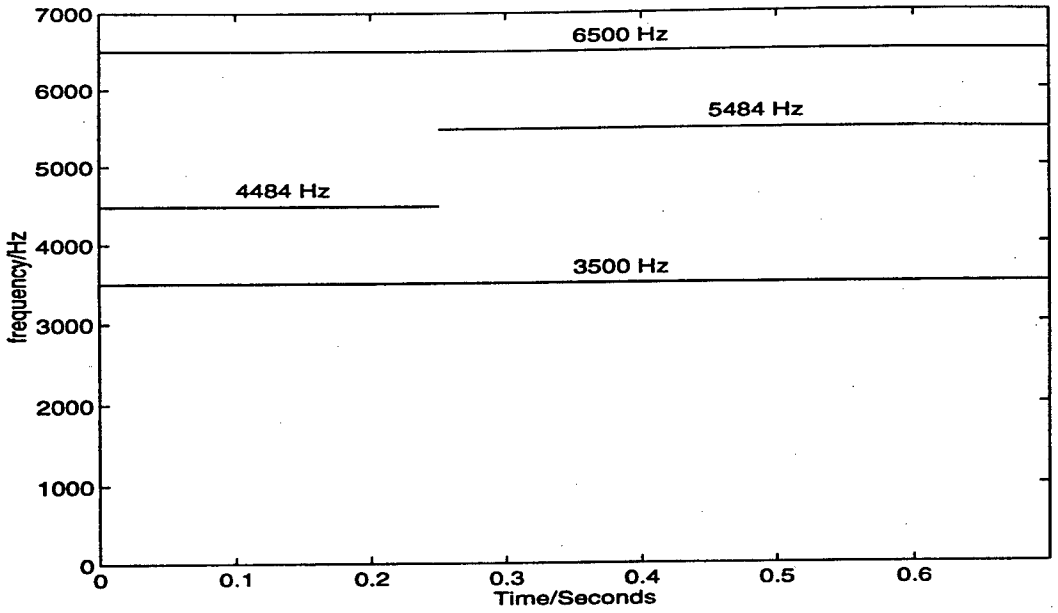


Figure 8.5: Estimated Time-Frequency Track of Two ON/OFF and Two CW Jamming Signals generated by IPUS for Scenario-2

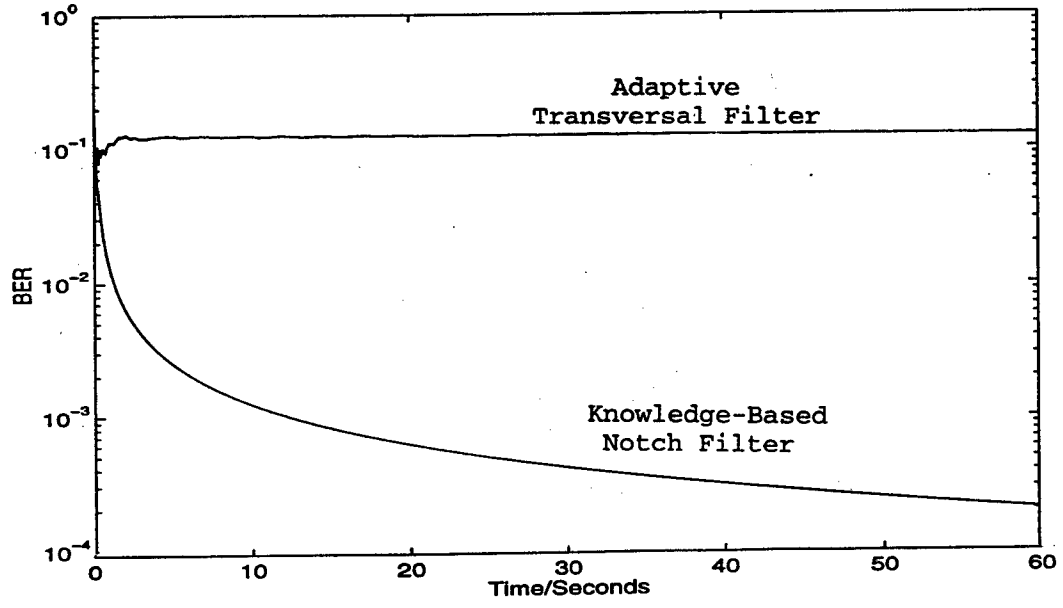


Figure 8.6: Performance Comparison for Scenario-2

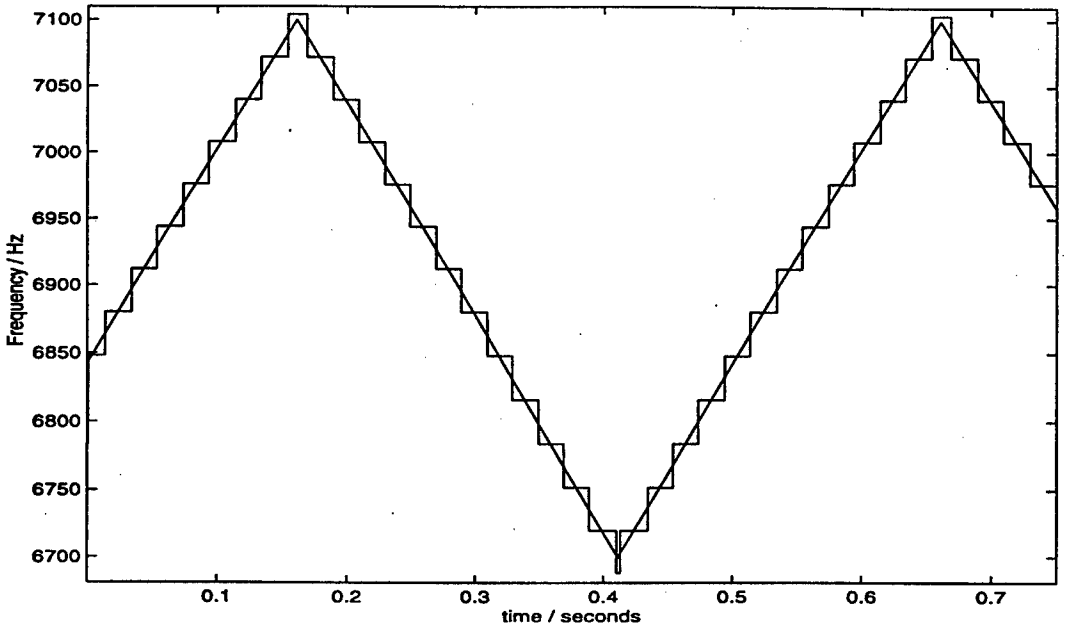


Figure 8.7: Actual and Estimated Time-Frequency Track of One LFM Jammer for Scenario-3

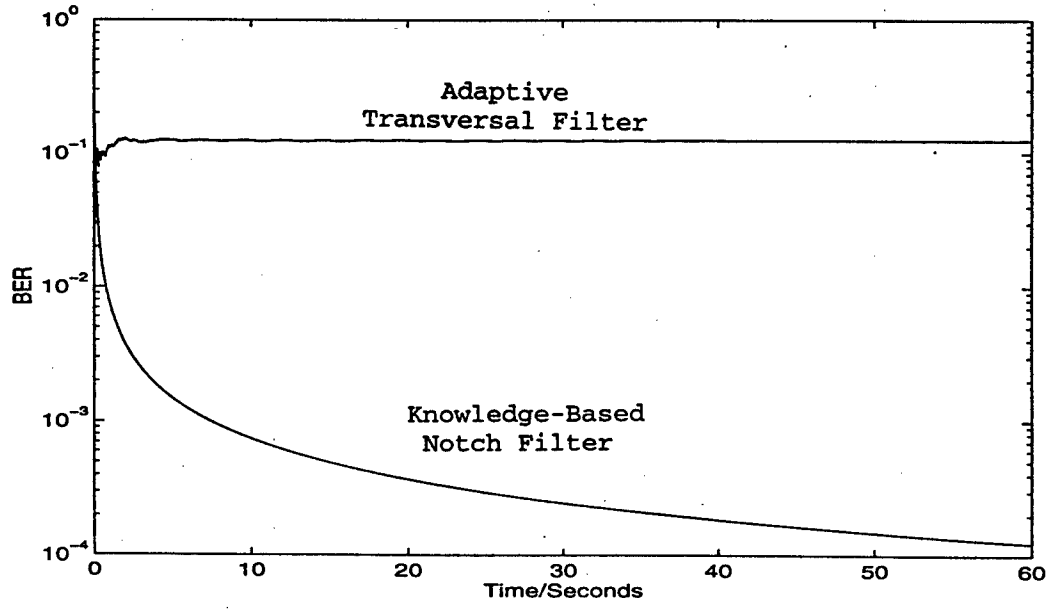


Figure 8.8: Performance Comparison for Scenario-3

Chapter 9

SUMMARY AND DISCUSSION

9.1 Summary

Communication in military environments presents quite a challenge because the interference that affects the communication could be intentional and hostile. Another important problem is that transmitted information may be intercepted by unintentional users. To provide satisfactory communication system performance, sophisticated signal processing techniques are required both to mitigate the effects of intentional interference and to lower the probability of interception. Advanced modulation schemes employing spread-spectrum and coding techniques are commonly used to decrease system vulnerability to interference and to lower its probability of interception. In many cases, even these techniques are not able to yield the desired level of performance and additional interference rejection techniques need to be employed. A significant number of interference cancellation schemes have been introduced in the literature. While these schemes are able to provide effective cancellation for specific types of interference, there is no single interference cancellation scheme that is able to effectively suppress all types of interference.

In this report we presented a novel knowledge-based interference cancellation framework for spread-spectrum communication systems. This innovative knowledge-based interference

cancellation technique utilizes IPUS, an expert system for the Integrated Processing and Understanding of Signals, to monitor the communication signal environment in order to determine the parameters of interfering signals within a prespecified accuracy. After having monitored the environment, the knowledge- based interference cancellation technique selects the most suitable cancellation filter from a library of cancellation filter banks in order to eliminate the interference signal from the received signal.

In this effort we limited our attention to direct-sequence spread-spectrum communication systems and to the set of jammers consisting of continuous wave, linear FM and ON/OFF jammers. We evaluated the performance of our schemes using computer-aided modeling and simulation techniques.

In addition, extensive experimentation was carried out in the development of rules for selection of the most suitable interference cancellation schemes. A number of interference scenarios were investigated and results demonstrated the superior performance of our knowledge-based interference cancellation schemes.

9.2 Discussion and Suggestions for Future Work

The goal of this research project was to provide a proof-of-concept demonstration for our knowledge-based interference cancellation framework. Only a limited set of interferers were considered in this first phase of the research project. In the next phase, we shall introduce additional jammers to our set of allowable jammers such as broad-band jammers, hopping jammers and intelligent jammers. This will require extension of $IPUS$ capabilities to be able to handle new classes of jammers, development of filter section rules, and inclusion of additional interference cancellation schemes to the library of interference cancellers.

During our protype development, ideal conditions were assumed, For example, perfect synchronization between the transmitter and the receiver was assumed. Also, no fading or multipath on the communication channel was assumed. In the next phase of the effort, we shall introduce these and other sources of uncertainty. Performance of our knowledge-

based scheme in the presence of non-ideal conditions will be investigated and compared with other interference cancellation algorithms. Thus, robustness of our schemes will be examined and methods for improving robustness will be determined.

Another interesting problem is the development of knowledge-based adaptive filtering for interference cancellation. One can investigate the development of reduced order adaptive transversal filters based on the interference characteristics available from $IPUS$. It is expected that such filters will be much simpler to implement while maintaining their adaptive behavior in case the information available from $IPUS$ is not accurate enough.

Our research in the first phase concentrated on knowledge-based interference cancellation in the time-domain. Knowledge-based interference cancellation utilizing transform-domain processing remains to be investigated. The interesting problem then arises of how to utilize the information available from $IPUS$ most effectively with regard to both time-domain and in Transform-domain processing.

Thus far, the research has emphasized a knowledge-based approach to interference cancellation. However, such propagation characteristics of the transmission medium, as fading, multipath and additive/multiplicative disturbances should be taken into account in the operation of a reliable communications link. In addition to the signal processing at the receiver, these factors can influence transmitter choices dealing with modulation type, data rate, and transmitted power. It remains to explore how $IPUS$ can be used to characterize the transmission channel and how this information can be employed by both the transmitter and receiver to improve communications performance.

Finally, with the newer version of SPW, it would be possible to embed $IPUS$ within SPW. It remains to be investigated whether it would be an effective architecture or the two should be on different machines to allow for distributed computing.

Bibliography

[1] S. Haykin, *Digital Communications,* Wiley, 1988.

[2] R. C. Dixon, *Spread Spectrum Systems,* Wiley-Interscience, 1976.

[3] R. L. Pickholtz, D. L. Schilling, and L. B. Milstein, "Theory of Spread-Spectrum Communications," *IEEE Trans. on Commun.,* vol. COM-30, pp. 885-884, May 1982.

[4] L. B. Milstein, " Interference Rejection Techniques in Spread-Spectrum Communications," *Proceedings of The IEEE,* vol. 76, no. 6, pp. 657-671 June 1988.

[5] J. D. Laster, and J. H. Reed, "Interference Rejection in Digital Wireless Communication," *IEEE Signal Processing Magazine,* pp.37-62, May 1997.

[6] L. B. Milstein, and P. Das, "An Analysis of a Real-Time Transform Domain Filtering Digital Communication System, Part 1: Narrowband Interference Rejection," *IEEE Trans. Commun.,* Vol. COM-28, pp. 816-824, June 1980.

[7] L. B. Milstein, and P. Das, "Spread-Spectrum Receiver Using Acoustic Surface Wave Technology'" *IEEE Trans. on Commun.,* vol. COM-25, No. 8, pp. 841-847, Aug. 1977.

[8] H. V. Poor, and L. A. Rusch, " Narrowband Interference Suppression in Spread-Spectrum CDMA," *IEEE Personal Communications Magazine,* pp. 14-27, Third Quarter 1994.

[9] V. R. Lesser, S. H. Nawab, and F. I. Klassner, "IPUS 1 : An Architecture for the Integrated Processing and Understanding of Signals," *Artificial Intelligence*, vol. 77, pp 129-171, 1995.

[10] S. H. Nawab and T. F. Quatieri, "Short-Time Fourier Transform," in *Advanced Topics in Signal Processing* Edited by Jae S-Lim and A.V.Oppenheim, Prentice Hall, Englewood Cliffs, N.J., 1988.

[11] F. Klassner, "Data Reprocessing in Signal Understanding Systems," *Doctoral dissertation*, University of Massachusetts, Amherst, September 1996.

[12] Y. Bar-Shalom, and T. E. Fortmann, *Tracking and Data Association,* Academic Press, 1988.

[13] D. B. Reid, "An Algorithm for Tracking Multiple Targets," *IEEE Trans. on Automatic Control,* vol. AC-24, no. 6, pp. 843-54, 1979.

[14] R. A. Singer, "Estimating Optimal Tracking Filter Performance for Manned Maneuvering Targets," *IEEE Trans. on Aerospace and Electronic Systems*, vol. AES-6, no. 4, pp. 473-83, 1970.

[15] E. Kreyszig, *Advanced Engineering Mathematics,* Seventh Edition, Wiley, 1993.

[16] F. J. Harris, "On the Use of Windows for Harmonic Analysis with the Discrete-Fourier Transform," *Proceedings of the IEEE*, vol. 66, no. 1, pp. 51-83, 1995.

[17] J. M. Winograd and S. H. Nawab, "A C++ Software Environment for the Development of Embedded Signal Processing Systems," *Proc. ICASSP 1995*, vol. 4, pp. 2715-19, Detroit, May 1995.

*U.S. GOVERNMENT PRINTING OFFICE: 1999-610-130-81110

MISSION OF AFRL/INFORMATION DIRECTORATE (IF)

The advancement and application of information systems science and technology for aerospace command and control and its transition to air, space, and ground systems to meet customer needs in the areas of Global Awareness, Dynamic Planning and Execution, and Global Information Exchange is the focus of this AFRL organization. The directorate's areas of investigation include a broad spectrum of information and fusion, communication, collaborative environment and modeling and simulation, defensive information warfare, and intelligent information systems technologies.